

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**TIAGO BEHENCK DOS SANTOS**

**MONITORAMENTO DE COLMEIA POR MEIO DE INTERNET DAS COISAS**  
**USANDO A UMA APLICAÇÃO MOBILE**

**CRICIÚMA**

**2019**

**TIAGO BEHENCK DOS SANTOS**

**MONITORAMENTO DE COLMEIA POR MEIO DE INTERNET DAS COISAS  
USANDO A UMA APLICAÇÃO MOBILE**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Me. Paulo João Martins

**CRICIÚMA**

**2019**

TIAGO BEHENCK DOS SANTOS

**MONITORAMENTO DE COLMEIA POR MEIO DE INTERNET DAS COISAS  
USANDO UMA APLICAÇÃO MOBILE**

Trabalho de Conclusão de Curso aprovado  
pela Banca Examinadora para obtenção do  
Grau de Bacharel, no Curso de Ciência da  
Computação da Universidade do Extremo  
Sul Catarinense, UNESC, com Linha de  
Pesquisa em Comunicação de Dados

Criciúma, 03 de Dezembro de 2019.

**BANCA EXAMINADORA**



Prof. Me. Paulo João Martins – UNESC – Orientador



Prof. Me. Luciano Antunes – UNESC



Prof. Esp. Marcel Campos Inocêncio – UNESC

**Dedico este trabalho aos meus pais, que são  
minha fonte de inspiração e que não  
mediram esforços para que eu chegasse até  
essa etapa da minha vida.**

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus, por proporcionar inúmeras bênçãos e fornecer a sabedoria necessária para percorrer os caminhos corretos, afim de alcançar meus objetivos e concluir mais esta etapa da minha vida.

Aos meu pais, Valmir Silva dos Santos e Sandra Behenck dos Santos, que sempre me apoiaram em todos os momentos, não mediram esforços diante os desafios que enfrentei, me orientando e mostrando o quanto é importante batalhar por algo que se deseja. A eles todo o meu amor e agradecimento.

A minha namorada Lithielle Goulardt da Silva, que sempre esteve ao meu lado, sendo paciente e nunca deixou de acreditar no meu potencial, me dando forças fazendo com que eu persistisse na luta para alcançar os meus objetivos. A ela, todo meu amor e carinho.

Agradeço ao meu amigo João Paulo Niehues Wessler que esteve ao meu lado durante esta jornada da graduação, onde juntos superamos barreiras com o apoio um do outro. A ele minha eterna gratidão e que essa amizade permaneça para a vida toda.

Um agradecimento especial para o Vinícius Magnus que não mediu esforços para me ajudar no desenvolvimento do protótipo, seu auxílio foi fundamental para sanar as minhas dúvidas técnicas. A ele minha gratidão e um agradecimento especial.

A todos os meus amigos em geral, que me deram forças e apoio para a conclusão desta etapa, sou muito grato a todos vocês.

Um agradecimento especial ao meu orientador Paulo João Martins, que não mediu esforços para me auxiliar em tudo o que fosse necessário para que eu conseguisse obter sucesso nesta etapa. A ele meu eterno agradecimento.

A UNESC e a todos os professores do curso de Ciência da Computação, que proporcionaram um ótimo ambiente para o aprendizado, bem como profissionais qualificados e dedicados na missão de repassar os seus conhecimentos a todos os alunos. A todos vocês meu muito obrigado.

**“Não há nada que dominemos inteiramente a  
não ser os nossos pensamentos.”**

**René Descartes**

## RESUMO

O emprego de tecnologias em diversas áreas da produção agrícola é uma realidade brasileira. No ramo da apicultura o cenário não se difere, grandes produtores têm acesso a ferramentas tecnológicas buscando otimizar a produtividade. Porém o acesso a meios tecnológicos não é uma realidade unânime na agricultura, como na produção de mel, por exemplo. Para implantar um sistema de monitoramento em uma colmeia existem um custo referente aos equipamentos empregados, o que acaba se tornado caro e dificultando a aquisição para o pequeno e médio produtor. Diante deste contexto, o presente trabalho objetiva a pesquisa e o desenvolvimento de uma solução de baixo custo para o monitoramento das variáveis de temperatura, umidade relativa do ar, peso e ruído em uma colmeia. Para tal foi desenvolvido um protótipo, aplicando os conceitos de Internet da Coisas e utilizando-se a placa de prototipação Arduino integrado a sensores específicos para cada variável monitorada. Os dados são coletados e enviados para um servidor na nuvem. Posteriormente, para o gerenciamento dessas informações, foi realizado o desenvolvimento de uma aplicação mobile, utilizando-se do framework React Native, onde o apicultor tem a possibilidade de visualizar e também efetuar uma pesquisa dentro de intervalo de data desejado a respeito dos dados coletado. Por fim, obteve-se um protótipo de sistema de monitoramento de colmeia, que pode ser aplicado a um ambiente real de produção, de modo a auxiliar na produção de mel e derivados e gerar melhores índices de produtividade

**Palavras-chave:** Apicultura. Monitoramento. Microcontroladores. Internet das Coisas. *React Native*.

## **ABSTRACT**

It is notorious that the use of technology is being more present in people's daily lives and in a level of participation being exponentially used in production environments, which progressively provides improvements and optimizes all types of activity. In the scenario of pig production is no different, where the continuous technological implementation, provides increased productivity and assistance in the management of animals. However, unfortunately this reality is not present in all the locals productions in the Brazilian territory, with this, still having much to improve mainly in small and medium producing properties. In this context, the present work aims at the research and development of a prototype for the monitoring of variables of temperature, humidity, sound frequency and luminosity related to animal welfare levels in swine. For that, a simulation was build in a model, using the Arduino microcontroller integrated with sensors specific to each monitored variable, through which the data is collected and sent to a server in the cloud. For the management of the information, was development a mobile application using the framework Ionic, which the user can manage all the data flow obtained and control the climate in the environment by the activation of equipment coupled to the model, simulating the cooling and heating. Finally, a prototype of a monitoring system was obtained, which can be applied to a pig production environment, in order to optimize management and generate better productivity indexes.

**Keywords:** Swine. Animal welfare. Microcontrollers. Monitoring. Framework Ionic.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Colmeia de Schenk .....	19
Figura 2 – Colmeia de Langstroth .....	19
Figura 3 – Demonstração da intersecção da computação ubíqua. ....	25
Figura 4 – Camadas Android.....	29
Figura 5 – Camadas da arquitetura iOS.....	31
Figura 6 – Funcionamento da bridge no React Native .....	36
Figura 7 – Arduino UNO Rev 3.....	39
Figura 8 – Comparativo entre as opções de Arduino .....	40
Figura 9 – Sensor AM2302 DHT22 .....	41
Figura 10 – Sensor PT 100 .....	42
Figura 11 – Sensor HDC1080 .....	42
Figura 12 – Sensor LM35 .....	43
Figura 13 – Sensor de ruído KY-038.....	43
Figura 14 – Sensor de peso .....	44
Figura 15 – Módulo HX711.....	45
Figura 16 – Módulo HC-05 .....	46
Figura 17 – Módulo Wi-Fi ESP8622 .....	47
Figura 18 – Módulo Wi-Fi ESP32.....	47
Figura 19 – Arquitetura da aplicação.....	56
Figura 20 – Tabelas do banco de dados .....	58
Figura 21 – Estrutura do DHT22 .....	60
Figura 22 – Estrutura sensor KY-038.....	61
Figura 23 – Adaptador ESP8266 ESP-01 .....	62
Figura 24 – Sensor de peso .....	63
Figura 25 – Módulo conversor HX711 .....	64
Figura 26 – Resumo dos recursos do Arduino Mega2560 .....	65
Figura 27 – Tabelas do banco de dados .....	66
Figura 28 – Comando usado para criar a tabela apicultor.....	66
Figura 29 – Selecionando dados da tabela apicultor.....	67
Figura 30 – Conexão banco de dados .....	67
Figura 31 – Comando para buscar registro .....	68
Figura 32 – Consulta tabela .....	68

Figura 34 – Dados vindo da API.....	71
Figura 35 – QR Code gerado .....	72
Figura 36 – Tela menu principal .....	73
Figura 37 – Tela dos dados dos sensores.....	74
Figura 38 – Tela de histórico dos sensores.....	75
Figura 39 – Configuração de mínimo e máximo dos sensores.....	76
Figura 40 – Tela de histórico dos sensores.....	77
Figura 41 – Tabela leituras com os dados.....	79
Figura 42 – Gráfico da variação de temperatura .....	81
Figura 43 – Gráfico da variação da umidade.....	81
Figura 44 – Gráfico da variação da umidade.....	82
Figura 45 – Gráfico da variação da umidade.....	83

## LISTA DE TABELAS

Tabela 1 – Ranking de municípios produtores de mel de Santa Catarina.....	15
Tabela 2 – Ciclo até atingir a fase adulta .....	21
Tabela 3 – Sistemas de dispositivos móveis e suas respectivas características .....	32
Tabela 4 – Comparativo das possibilidades de desenvolvimento .....	33
Tabela 5 – Equipamentos utilizados para o desenvolvimento do protótipo .....	55
Figura 33 – Organização das pastas.....	70

## LISTA DE ABREVIATURAS E SIGLAS

°C	Graus(s) Celsius
AC	Corrente Alternada
ABEMEL	Associação Brasileira de Exportadores de Mel
CIRAM	Centro de Informações De Recursos Ambientais e de Hidrometeorologia de Santa Catarina
CSS	<i>Cascading Style Sheets</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
EPAGRI	Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina
FAASC	Federação das Associações de Apicultores e Meliponicultores de Santa Catarina
GHz	Gigahertz
GPRS	<i>General Packet Radio Services</i>
GSMA	<i>Global System for Mobile Communications</i>
HTML	<i>Hyper Text Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
JSON	<i>JavaScript Object Notation</i>
kB	<i>Kilobyte</i>
PC	<i>Personal Computer</i>
RAM	<i>Random Access Memory</i>
RFID	<i>Radio-Frequency Identification</i>
SDK	<i>Software Development Kit</i>
SO	Sistema Operacional
SRAM	Memória de Acesso Randômico Estática
URL	<i>Uniform Resource Locator</i>
USB	<i>Universal Serial Bus</i>
V	Volts

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>6</b>
1.1 OBJETIVOS.....	7
1.1.1 Objetivo Geral.....	7
1.1.2 Objetivos Específicos .....	8
1.2 JUSTIFICATIVA .....	8
1.3 ESTRUTURA DO TRABALHO .....	10
<b>2 BASE APÍCOLA .....</b>	<b>11</b>
2.1 APICULTURA E IMPORTÂNCIA DAS ABELHAS NA NATUREZA .....	11
2.2 UM HISTÓRICO DA APICULTURA NO BRASIL .....	13
2.3 APICULTURA DE PRECISÃO .....	15
2.4 O APIÁRIO .....	17
2.4.1 Organização de uma colmeia .....	20
2.4.2 Termorregulação .....	22
<b>3 TECNOLOGIAS PARA O DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>24</b>
3.1 INTERNET DAS COISAS.....	24
3.2 PLATAFORMAS MÓVEIS.....	26
3.2.1 Plataforma Android .....	27
3.2.2 Plataforma iOS.....	30
3.3 APLICAÇÕES MÓVEIS.....	32
3.3.1 Aplicações Nativas.....	33
3.3.2 Aplicações Híbridas .....	34
3.3.2.1 React Native.....	35
3.3.3 Aplicações WebApps .....	36
<b>4 MICROCONTROLADOR E SENSORES .....</b>	<b>38</b>
4.1 ARDUINO.....	38
4.2 SENSORES DE TEMPERATURA E UMIDADE .....	41
4.2 SENSORES DE SOM .....	43
4.3 SENSORES DE PESO.....	44
4.4 CONEXÕES COM DISPOSITIVOS EXTERNOS .....	45
4.4.1 Módulo Bluetooth .....	45
4.4.2 Módulo Wifi .....	46
<b>5 TRABALHOS CORRELATOS.....</b>	<b>48</b>

5.1 WEBBEE .....	48
5.2 BIG BROTHER FOR BEES (3B) – ENERGY NEUTRAL PLATAFORM FOR REMOTE MONITORING OF BEEHIVE IMAGERY AND SOUND .....	48
5.3 BEEHIVE MONITOR .....	49
5.4 CONTINUOUS MONITORING OF BEEHIVES’ SOUND FOR ENVIRONMENTAL POLLUTION CONTROL.....	49
5.5 BEE HIVE TEMPERATURE AND SOUND MONITOR.....	49
<b>6 MONITORAMENTO DE COLMEIAS POR MEIO DE INTERNET DAS COISAS CONECTADO A UMA APLICAÇÃO MOBILE .....</b>	<b>51</b>
6.1 METODOLOGIA.....	51
6.1.1 Ferramentas e recursos utilizados .....	53
6.1.2 Arquitetura da aplicação.....	56
6.1.3 Diagrama de classes .....	58
6.1.4 Sensoriamento e processamento do Arduino .....	59
6.1.5 Ambiente de processamento.....	65
6.1.6 Desenvolvimento da aplicação móvel .....	69
6.1.7 Aplicação em funcionamento.....	73
6.1.8 Testes do projeto.....	77
6.2 RESULTADOS E DISCUSSÕES .....	80
<b>7 CONCLUSÃO .....</b>	<b>85</b>
<b>REFERÊNCIAS.....</b>	<b>87</b>

## 1 INTRODUÇÃO

Entende-se por cultura ou exploração das abelhas, ou ainda apicultura, o ramo de agricultura que tem por fim explorar esses insetos e tirar deles o máximo e o melhor rendimento possível (D'ALMEIDA, 1983). As abelhas têm um papel fundamental no ecossistema. Conforme Lima (1979), seriam grandes as dificuldades de alimentação da humanidade sem a existência das abelhas, pois elas são os principais insetos polinizadores que existem e sem elas milhares de espécies de plantas teriam deixado de existir. Além da sua função ecológica, as abelhas e seus produtos demonstram um grande potencial econômico.

No ano de 2015 o faturamento do setor apícola no Brasil atingiu a marca de R\$ 358,8 milhões (IBGE, 2016). Ele também tem grande expressão quando se trata de exportação, ao se considerar que em dezembro de 2016 foram exportadas mais de 2 mil toneladas de mel (ALIECEWEB, 2017).

A colheita de mel se trata de uma atividade que requer esforço físico acentuado causando um desgaste para o apicultor, uma vez que o peso das colmeias cheias é considerável (CAMARGO et al., 2002). A forma de verificação das colmeias por vezes pode se mostrar invasiva, prejudicando a produção de mel. Tratando-se de um grande produtor, pode acarretar prejuízos, além do custo da mão de obra qualificada para exercer determinada função. Visando estes pontos demonstra-se viável o desenvolvimento de um sistema que auxilie nas averiguações periódicas de algumas variáveis (umidade, temperatura, peso e ruído) simplificando o processo de aferição manual que será apenas a retirada do mel.

O governo de Santa Catarina monitora algumas colmeias, aferindo temperatura e peso, por meio de um sistema desenvolvido pela EPAGRI/CIRAM. Porém o acesso a este tipo de equipamento é restrito devido aos seus altos custos.

Empregando o conceito atual de Internet das Coisas, também conhecida por *Internet of Things* (IoT), e possui uma enorme aplicabilidade para o desenvolvimento do protótipo. O sistema pode se integrar a uma plataforma, seja ela um aplicativo ou um *site*, de uma forma dinâmica, trazendo os resultados obtidos pelos sensores de maneira simples e eficaz ao usuário. Utilizando sensores de peso, que permita fazer a pesagem de uma “casa de mel”, sem a necessidade de aferir de forma manual, diminui o desgaste físico do apicultor e também reduz o estresse das abelhas.

Para prototipação, existe o sistema embarcado denominado Arduino, que é uma plataforma de prototipagem eletrônica *open source* (ARDUINO, 2019, tradução nossa). Baseia-se em *hardware* e *software* flexíveis e é aplicado como ferramenta principal para o processamento e gerenciamento dos sensores. Sua aplicabilidade e integração com inúmeras opções de programação aliado ao seu custo acessível. Com todos os dados coletados da colmeia em questão, pretende-se desenvolver um aplicativo com o intuito de exibir estas informações. Planejou-se criar um aplicativo *mobile* devido a popularidade do *smartphone*, contudo há alguns pontos a serem observados, como por exemplo, a existência de colmeias em um terreno isolado, onde inexistente o sinal de telefone ou Internet móvel.

Levando em consideração a importância das abelhas, no contexto da natureza e indo além das questões ambientais, como por exemplo, o quesito econômico onde os pequenos e os grandes produtores encontram dificuldades para obter um monitoramento de precisão que futuramente resulte em uma produção maior de mel, por exemplo. Com esta situação, pretende-se desenvolver um sistema que tem a capacidade de monitorar variáveis (peso, umidade, ruído e temperatura) por meio de sensores, ligados a um aplicativo utilizando Internet das Coisas e exibindo os dados de maneira fácil para o produtor.

Com este sistema há possibilidade de reduzir os custos a longo prazo de extração de mel, além de ter um maior controle quanto a saúde da colmeia, bem como ter dados confiáveis para um melhor planejamento e controle da produção.

## 1.1 OBJETIVOS

A seguir, salientam-se os objetivos do projeto de pesquisa.

### 1.1.1 Objetivo Geral

Aplicar uma solução de monitoramento de variáveis em colmeias, aplicando o conceito de Internet das Coisas e integrando os dados por meio de uma aplicação móvel.



### 1.1.2 Objetivos Específicos

Tem-se os seguintes objetivos específicos do projeto de pesquisa:

- a) Descrever os conceitos e características da apicultura;
- b) Utilizar os conceitos de Internet das coisas para a comunicação hardware e software;
- c) Empregar sensores de prototipação para peso, umidade, temperatura e ruído;
- d) Implementar um aplicativo para visualização/consulta dos dados.

### 1.2 JUSTIFICATIVA

A tecnologia já faz parte do dia a dia da maioria das pessoas, mas não só presente em tarefas cotidianas, bem como está em setores importantes da economia, indústria, saúde. Ela vem auxiliando em diversas áreas como por exemplo no ramo da apicultura, que é uma derivação da área da agricultura.

A apicultura também desperta interesse em diversos segmentos da sociedade pela sua capacidade de gerar empregos no campo, possibilidade de obtenção de bons lucros (BROWN; PAXTON, 2009). O setor de apicultura brasileiro tem uma enorme expressão no mercado, onde além dos lucros por meio da exportação, também concede emprego no campo. Porém a mão de obra deve ser qualificada para o manuseio das colmeias no momento da extração do mel.

Logo uma nova tecnologia que auxilie neste gerenciamento da colmeia se mostra muito útil a um produtor de mel na qual iria evidenciar quesitos para maximizar a sua produção e por consequência lógica acarretando maiores lucros. Um sistema que possibilite ter um controle de peso, umidade, temperatura e ruído de uma colmeia, por exemplo.

Para esta integração do meio físico com a parte lógica é comum se utilizar de hardware práticos. Arduino é uma plataforma para prototipagem de objetos interativos usando a eletrônica (MELLIS et al., 2007) o qual se mostra muito viável quando se trata de acoplar sensores e utilizar como principal ferramenta de gerenciamentos dos mesmos.

Aplicando este modelo de prototipação precisa-se de uma ligação para a solução em questão onde os dados coletados por meios dos sensores têm como destino final uma aplicação *mobile* e a exibição dos resultados obtidos. A Internet das coisas (IoT – *Internet of things*) trata de uma integração entre um objeto físico acoplado a um sistema que tem acesso à Internet, assim permitindo que todos que estão diante da mesma rede possam trocar e armazenar informações do meio. Segundo Deidmar et al. (2017), é um conceito em que todos os objetos do cotidiano estariam conectados à Internet, agindo de modo inteligente e sensorial. Consiste na ideia da fusão do “mundo real” com o “mundo digital”, fazendo com que o indivíduo possa estar em constante comunicação e interação, seja com outras pessoas ou objetos.

O trabalho tem intuito de desenvolver um protótipo que possa fazer essas averiguações e retornar os valores em um aplicativo móvel. Uma vez que o *smartphone* se tornou um objeto fundamental no nosso dia a dia. Atualmente, existem cerca de 1,5 bilhões de computadores habilitados para acesso à Internet e mais 1 bilhão de celulares com acesso à Internet. A atual “Internet dos PCs” irá avançar para uma “Internet das coisas” em que 50 a 100 bilhões de dispositivos serão conectados a Internet até 2020 (SANTUCCI, 2010).

Um relatório da *Global System for Mobile Communications* (GSMA), entidade que representa operadoras móveis do mundo, aponta o Brasil com o país com mais *smartphones* conectados à Internet na América Latina. São 234,6 milhões de conexões sem fio no país no terceiro trimestre do ano (2017), sendo 73% a partir desses aparelhos, 35% usando tecnologia 4G. (AMPUDIA, 2017).

Logo, uma aplicação móvel se mostra de grande vantagem e de fácil acesso. O sistema implementado irá agregar para a produção de mel, facilitando as verificações que até então se mostram manuais, o que é um tanto rudimentar, além da informação que estará disponível a qualquer momento para o produtor. Desta forma, pode-se minimizar os custos de produção e também reduzir a interferência nas colmeias, pois com a obtenção dos dados não será mais necessário o manuseio antes de que determinados parâmetros sejam registrados.

### 1.3 ESTRUTURA DO TRABALHO

O vigente trabalho é contemplado por seis capítulos, sendo o primeiro capítulo constituído de introdução, onde se encontra uma abordagem sucinta e concisa, levando como premissas os objetivos específicos, bem como o geral. A seção também contém a justificativa e a estrutura do trabalho.

O segundo capítulo aborda um amplo panorama da base apícola, salientando questões biológicas das abelhas e de assuntos referentes à apicultura de precisão.

O terceiro capítulo descreve a respeito das tecnologias que serão aplicadas no protótipo, discorrendo sobre plataformas móveis e seus aplicativos, além de caracterizar pontos referentes à Internet da Coisas.

No quarto capítulo aborda-se microcontroladores, sensores e dispositivos de conexão passíveis de serem utilizados no desenvolvimento do protótipo.

Os trabalhos correlatos estudados para a confecção desse trabalho estão resumidos no capítulo quinto.

O sexto capítulo descreve a metodologia do trabalho proposto, descrevendo os procedimentos referente ao protótipo desenvolvido. Também neste capítulo, contém os resultados obtidos com o projeto.

No sétimo capítulo aborda-se a conclusão alcançada com a pesquisa e desenvolvimento do projeto, além de sugestões de trabalhos futuros

E por fim encontra-se as referências bibliográficas utilizadas no desenvolvimento do trabalho.

## 2 BASE APÍCOLA

Nesta seção que começa é apresentada uma visão geral sobre os conceitos da apicultura e sua natureza, e os mesmos são levados como base neste projeto de pesquisa. Contém questões biológicas das abelhas, ligadas ao manejo das colmeias, econômicas além da apicultura de precisão.

### 2.1 APICULTURA E IMPORTÂNCIA DAS ABELHAS NA NATUREZA

Entende-se por cultura ou exploração das abelhas, ou ainda apicultura, o ramo da agricultura que tem por fim explorar estes insetos e tirar deles o máximo e o melhor rendimento possível (D'ALMEIDA, 1983). E não se trata de uma atividade que é recente. A abelha é um inseto milenar e está presente em toda a história da humanidade desde os tempos mais remotos, como por exemplo nos fósseis descobertos bem como nos hieróglifos dos monumentos egípcios. Vestígios encontrados em uma pirâmide, junto ao túmulo misterioso, se infere que o mel foi certamente uma iguaria apreciada no tempo de Cleópatra (WIESE, 1985).

Na idade média desenvolveram-se técnicas aperfeiçoadas na extração do mel, sem a perdas das abelhas. Então surgiu a técnica de aplicar fumaça dentro das colmeias. Com a fumaça as abelhas iam para o fundo, incluindo a rainha, e posteriormente eram retirados somente os favos da frente, deixando uma reserva para as abelhas (BERALDO, 2011).

Com o decorrer do tempo novas técnicas foram surgindo no que diz respeito ao manejo das abelhas e na extração do mel. Somente em 1851 o americano Lorenzo Lorraine Langstroth, conhecido a nível mundial no ramo apícola por Langstroth, descobriu o “espaço-abelha”. A área de desenvolvimento conhecida como espaço-abelha permite que as abelhas se movimentem, produzam mel e cuidem das mais jovens. De acordo com as suas teorias, a quantidade ideal de espaço entre as camadas de favo de mel é de 64 a 79 milímetros (WILSON, 2001).

A apicultura se transformou numa atividade mundialmente lucrativa na qual pode ser praticada pelo produtor rural ou agricultor familiar obtendo bons resultados. E tem muitas vantagens, como: não exige grandes dimensões territoriais e além do

mel, as abelhas oferecem outros produtos/derivados que podem ser consumidos ou comercializados (EMBRAPA, 2007).

Se trata de animais de grande importância visando estabelecer o equilíbrio do ecossistema global. Seres que contribuem fortemente para a preservação da vida vegetal e também na manutenção da variabilidade genética. Sua principal função está relacionada diretamente na participação ativa do processo de polinização de diversas plantas existentes no planeta. (DUTRA, 2016).

Segundo Schirmer (1985), havendo uma polinização por intermédio das abelhas, fica garantida a fecundação de todas as flores, os frutos serão maiores e de melhor qualidade, com isso, na maioria das vezes, uma mesma área cultivada pode produzir cerca de 50% a mais por meio da polinização feita pelas abelhas.

E além do fato primordial que as abelhas desempenham, também existe a possibilidade de exploração comercial. O cultivo delas é de grande vantagem em inúmeros pontos, como afirma D'Almeida (1983), apicultura se trata, talvez, de todas as áreas da indústria agrícola a que menos capital exige, menos cuidado para manter e que menos embaraços causa, devido a matéria prima existente sempre em maior ou menor abundância, sem grandes preocupações para o produtor.

Os principais produtos derivados da prática apícola são o própolis, a geleia real, o pólen, a cera, apitoxina além do mel. Este que possui qualidades nutricionais únicas e de um vasto mercado (OLIVEIRA; SEABRA, 2006). O mel é o produto de maior interesse para a maioria dos apicultores brasileiros, responsável por gerar a maior parte dos lucros dos apicultores. Segundo Vieira (1989), mel é um produto tipicamente açucarado, de sabor em geral agradável, de altos valores energéticos, alimentar e terapêutico é também usado em produtos de beleza. É produzido pelas abelhas a partir do néctar que é coletado diretamente das plantas.

Ainda conforme Vieira (1989), a cera se constitui de um produto impermeável à água, formada por ácidos graxos, principalmente os cerótico e palmítico. A cera é um produto fisiológico produzido dentro da colmeia pelas abelhas cerieras, que tem de 12 a 18 dias de existência. A matéria prima é o próprio mel e para realizar a tarefa, as cerieras utilizam seu corpo como ferramenta (WIESE, 1985).

Com tamanha capacidade na criação destes pequenos insetos, a apicultura foi crescendo e tomando grandes proporções na economia no setor da agricultura. Em um relatório obtido por meio do site da Associação Brasileira de Exportadores de Mel

(ABEMEL), há pesquisas visando à compreensão das propriedades do mel nacional, melhorando as técnicas de manejo, fortalecendo a cadeia produtiva como um todo além da comercialização nacional e internacional dos produtos derivados das abelhas. Isso demonstra o grande potencial de um sistema que faça aferições da colmeia, visando melhorar a produção. Em 2015 a exportação do mel chegou na casa de 22 mil toneladas e vendida a um preço médio de US\$ 3,68/kg (ABEMEL, 2015).

Abelhas são importantes polinizadores de culturas que desempenham um papel significativo no sistema ecológico e na safra industrial. As abelhas são conhecidas pelo seu nível de sociedade, além é claro do mel e seus derivados. Isto ocorre porque vivem de forma agrupada, a qual se caracteriza como uma comunidade, existindo uma nítida distribuição de trabalhos bem como responsabilidade dos indivíduos onde todos contribuem para um fim comum, a sobrevivência do grupo.

## 2.2 UM HISTÓRICO DA APICULTURA NO BRASIL

A atividade apícola no Brasil não é algo recente, segundo dados da EMBRAPA (2002) acredita-se que por volta de 1840, abelhas foram trazidas pelo Padre Antônio Carneiro. Provavelmente as subespécies *Apis mellifera mellifera* (abelha preta ou alemã) e *Apis mellifera cárnica* tenham sido as primeiras abelhas a chegarem em nosso país.

Segundo Schirmer (1985), o primeiro produtor que se instalou com uma apicultura móbil e pioneiro apicultor europeu a vir para o Brasil foi Frederico Augusto Hannemann, em Rio Pardo no Rio Grande do Sul no ano de 1853.

Nos primórdios da apicultura brasileira pode-se citar outros nomes além de Hannemann, como Emílio Schenk e Don Amaro Van Emelen. Todos tiveram grande importância na introdução da apicultura no cenário nacional. Schenk é autor da obra O Apicultor Brasileiro, uma literatura de suma importância onde foram feitas inúmeras edições. Segundo Muxfeld (1987), Schenk foi responsável por trazer as rainhas douradas da Alemanha e, posteriormente, foi buscar abelhas italianas aclimatadas nos Estados Unidos que são origens das célebres abelhas italianas que se encontram no Rio Grande do Sul atualmente. Hoje a colmeia móvel leva seu nome como homenagem, “colmeia Schenk”.

Don Amaro Van Emelen, belga, monge, trouxe abelhas italianas para Olinda, Pernambuco. Organizou a escola de Agricultura e Veterinária de Tapera. Editou cinco vezes a Cartilha do Apicultor Brasileiro, além de publicar artigos sobre “A criação de Rainhas” e também “Abelhas, Mel e Cera”, fora os inúmeros artigos publicados no Almanaque das Abelhas (SCHIRMER, 1985).

Como grande parte das abelhas foram trazidas de um outro continente, isso implica diretamente na mudança de seu habitat natural. Como relata Servelin (2008), as abelhas adaptaram-se bem ao clima do país e desenvolveram-se de forma rápida, produzindo ótimos resultados, principalmente por se tratarem de abelhas dóceis (*Apis Mellifera Mellifera*) e de fácil manejo. Segundo Martinez e Soares (2012), a diversidade de climas, junto com a possibilidade de direcionar a produção para outros ramos, fazem do Brasil um país promissor no campo internacional da apicultura.

O Estado de Santa Catarina é um dos grandes produtores de mel brasileiro. Segundo Servelin (2008), a apicultura catarinense desenvolveu-se porque dispõe de flora abundante e diversificada, proporcionando uma produção de mel de excelente qualidade. Como aponta uma pesquisa feita pela Federação das Associações de Apicultores e Meliponicultores de Santa Catarina (FAASC) com apoio da Fundação do Banco do Brasil e do Sebrae, Santa Catarina é o maior produtor de mel por quilômetro quadrado do país e o maior exportador do produto, além de ser considerado como um dos melhores do mundo. A FAASC realizou um levantamento em 2016 dos municípios produtores de mel e dispôs em uma ordem de produção de mel. Como demonstra a tabela 1, observa-se Içara em que consta em segundo lugar no estado.

Tabela 1 – Ranking de municípios produtores de mel de Santa Catarina

Município	Apicultores	Colmeias	Produção (kg)
Bom Retiro	132	12945	310680
Içara	74	14227	293076
Urubici	30	10465	214532
Santa Terezinha	45	9816	201228
Fraiburgo	58	10759	193500
São Bonifácio	114	6602	135341
São Joaquim	62	6490	133045
Anitápolis	99	4942	101311
Angelina	188	4783	98051
São Bento do Sul	52	4505	92352

Fonte: Adaptado de FAASC (2016).

## 2.3 APICULTURA DE PRECISÃO

Apicultura de precisão se trata de uma estratégia de gestão de apiário com base na monitorização das colônias de abelhas individuais para minimizar o consumo de recursos e maximizar a produtividade das abelhas. Tem como principal requisito a possibilidade de coletar e analisar dados sobre a colmeia de forma contínua, em tempo real e de forma automatizada. (ZACEPINS; STALIDZANS; MEITALOVS, 2012, tradução nossa).

Com base nestas informações, os apicultores podem tirar conclusões importantes e em caso de necessidade executar algumas ações adicionais. Além disso, o desenvolvimento de tecnologias de comunicação sem fios pode ser empregue para medir variáveis necessárias de uma colmeia (MEZQUIDA e MARTINEZ, 2009, tradução nossa).

Alguns dos sensores mais utilizados para o monitoramento são: temperatura, ruído, umidade relativa, peso, dióxido de carbono e imagens. Somente alguns destes serão utilizados no desenvolver o projeto.

Três fatores que estão conectados com a enxameação da população de abelhas adultas são a quantidade e distribuição do feromônio e o ambiente da colmeia. Todos esses fatores estão relacionados a uma combinação de som, temperatura e a



umidade da colmeia. Apicultores devem tomar cuidados por seus enxames desde o controle da enxameação (FERRARI et al., 2008, tradução nossa).

O peso de uma colmeia reflete-se na saúde e produtividade da colônia (FITZGERALD et al., 2015) é um dado de grande utilidade para o apicultor e também pode-se obter uma estimativa da produção da colmeia de forma individual. O peso de uma colmeia habitada (soma da caixa, melgueira, tela e abelha) devem ser monitorados para identificar os diferentes estados da colônia como a ocorrência do néctar durante a época de forrageamento (período entre o início e fim da floração) ou o ganho diário de néctar e até mesmo a diminuição do exame por meio do peso da colmeia (ZACEPINS; STALIDZANS; MEITALOVIS, 2012, tradução nossa).

Padrões sonoros podem ser usados como indicadores da saúde da comunidade de abelhas. Apenas o padrão global produzido pelo enxame é determinante na mudança de comportamento (PÉREZ et al, 2016). O monitoramento de vibrações sonoras pode ser aplicado com estimativa do comportamento da comunidade de abelhas, identificando padrões de zumbido do exame que indique o possível abandono em massa da colmeia (BENCSEK et al., 2011), ou possíveis acidentes com a colmeia através da detecção de movimentos bruscos (MURPHY et al., 2015).

Referindo-se a dados sonoros coletados em colmeias pode-se estudar os recursos de som correlacionados com a atividade dos insetos, temperatura e a umidade do ecossistema da colmeia. Quando as abelhas estão começando a enxamear, o som é aumentado em amplitude e frequência para 300 Hz e, ocasionalmente, uma mudança rápida ocorre de 150 Hz a 500 Hz (FERRARI et al., 2008, tradução nossa).

Por meio da umidade relativa podem ser detectados eventos ligados a saúde da colmeia como o aumento no consumo de alimentos ou possível abandono da colmeia (ALMEIDA, 2008). A umidade relativa no interior da colmeia é mantida por volta dos 40%. Monitorar esse valor pois, se elevar muito por consequência da evaporação do néctar, as abelhas operárias imediatamente provocarão uma corrente de ar para o interior da colmeia, na tentativa de diminuir a umidade (CAMARGO et al., 2002).

## 2.4 O APIÁRIO

O apiário sofrerá interferência durante toda a sua existência ligado a fatores do meio ambiente no qual está instalado, tais como ventos, umidade, temperatura, chuvas, pássaros predadores e insetos concorrentes, de forma que tais fatores são, em grande parte, responsáveis pelo progresso ou não do apiário (BERALDO, 2011).

De acordo com Pereira et al. (2003), os apiários podem ser classificados em dois tipos: os migratórios e os fixos. Um apiário fixo é caracterizado pela permanência das colmeias durante todo o ano em uma única área, onde as abelhas irão explorar as fontes florais disponíveis em seu raio de ação. Assim a escolha do local se torna fundamental visando garantir segurança para as abelhas em relação a pessoas e outros animais (PEREIRA et al., 2003).

O apiário migratório compartilha das mesmas características de um apiário fixo, entretanto as abelhas são deslocadas ao longo do ano para explorar diferentes locais com recursos florais abundantes. (PEREIRA et al., 2003).

Segundo Vieira (1989), o local onde se instalara o apiário é de grande importância porque por mais que há as melhores instalações, as melhores abelhas e um bom manejo, de nada adianta se não houver alimentos suficientes.

O local para a instalação do apiário deve ser bem definido antes da chegada das colônias para que, posteriormente, não haja necessidade de deslocamentos das colmeias, o que acarretaria grande perda de abelhas campeiras que sempre voltam ao local de origem da colmeia. As abelhas possuem, em média, um raio de ação que abrange cerca de 2,5 quilômetros em volta da colmeia e por isso se houver necessidade de mudar a colmeia de local, a recomendação é que o novo local esteja a uma distância superior a 2 quilômetros (BERALDO, 2011).

Para a definição estratégica de onde instalar o apiário, vários pontos podem ser citados como (PEREIRA et al., 2003 apud EMBRAPA, 2017):

- a) flora apícola: é caracterizada pelas espécies vegetais que possam fornecer néctar e/ou pólen que são os produtos essenciais para a manutenção das colônias e para a produção de mel. O conjunto dessas espécies é denominado pasto apícola ou pastagem apícola. Assim, o apicultor deve planificar a formação do pasto apícola antes mesmo da instalação do apiário, levando em consideração fatores como as

espécies de abelhas com as quais se trabalhará e os períodos de floradas;

- b) acesso: o local do apiário deve ser de fácil acesso, favorecendo ao máximo a aproximação de veículos ao local de instalação das colmeias, o que facilita acentuadamente o manejo, o transporte da produção e, eventualmente, das colmeias;
- c) topografia: o terreno do apiário deve ser plano e com frente limpa, evitando-se áreas elevadas, em virtude da ação negativa dos ventos fortes. Terrenos em declive dificultam o deslocamento do apicultor pelo apiário e, conseqüentemente, o manejo das colmeias, principalmente durante a colheita do mel.

Após escolhido o melhor local para a instalação do apiário, levando em conta todos os pontos que se mostram de grande importância para que o enxame prospere, há diferentes tipos de colmeias/estrutura para abrigar as abelhas. Existem colmeias Schirmer, colmeia ou caixa Schenck, colmeia Langstroth dentre inúmeras outras.

Relacionada a colmeia Schenck, se encontra difundida e em uso nos estados do Rio Grande do Sul, Santa Catarina, Paraná e algumas regiões de São Paulo. Se trata de uma colmeia de forma retangular para expansão vertical e de câmara quente, considerando-se a posição dos quadros que são transversais ao alvado (WIESE, 1985). A figura 1 ilustra uma colmeia completa que leva o nome de Schenck.

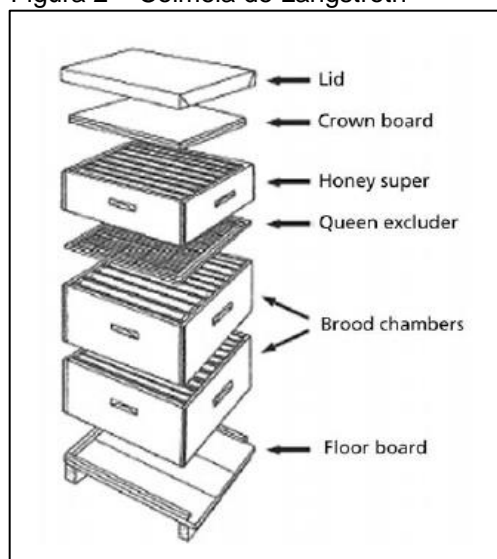
Figura 1 – Colmeia de Schenk



Fonte: Caixa de abelha (2018)

Porém atualmente o tipo de instalação utilizado em larga escala se trata da colmeia Langstroth. Em 1852, Lorenzo Lorain Langstroth, após longos estudos exaustivos da biologia das abelhas e que resultaria em uma descoberta a do “espaço abelha”. Graças a ele, hoje tem-se uma colmeia essencialmente funcional - que leva seu nome e também é conhecida por colmeia americana - com formato, dimensões, divisões, espaço útil e cores, na qual é de grande valia para o apicultor e excelente para as abelhas (VIEIRA, 1989). A figura 2 representa a colmeia de Langstroth.

Figura 2 – Colmeia de Langstroth



Fonte: Cramp (2008)

### 2.4.1 Organização de uma colmeia

As abelhas são insetos sociais, vivendo em colônias organizadas em que os indivíduos se dividem em castas, possuindo funções bem definidas que são executadas visando sempre à sobrevivência e manutenção do enxame (PEREIRA et al., 2003).

Dentro de uma colmeia é possível encontrar:

- a) rainha: responsável pela reprodução, é a única abelha da colmeia que se acasala com os machos (zangões) e pode pôr ovos que geram fêmeas (operárias e rainhas) e zangões;
- b) zangões: machos da colmeia, cuja única função é se acasalar com a rainha;
- c) operárias: realizam todo o trabalho da colmeia: coletam alimento (néctar e pólen) e água, cuidam das crias e da rainha, limpam a colmeia e defendem o enxame (EMBRAPA, 2007).

Dentre estas três divisões, as operárias que detém o maior número da população. Para manter uma colmeia saudável necessita de diversos cuidados, para lidar é necessário trabalhar com toda calma e atenção. Segundo Vieira (1989), quanto mais fraca a colmeia, mais mansas serão as abelhas e o oposto também, quanto mais forte mais agressivas elas se tornam. Conhecer de forma detalhada as atividades sociais das abelhas está ligado diretamente a sua produção de mel, afirma Schirmer (1985).

Na construção de seus depósitos, silos, armazéns, berços e dormitórios as abelhas adotam um padrão geométrico: o hexágono. Segundo Muxfeldt (1987), o hexágono é a figura mais apropriada e resistente, a qual pode ser construída com grande economia de material e de trabalho. Por trás destas simples construções se tem um grande feito na engenharia que levou cientistas a estudarem esses pequenos insetos voadores. A grossura normal de um favo varia de 22 a 23 milímetros, contando com uma inclinação de 4 a 5 graus, para que não escorra o mel, como relata Muxfeldt (1987).

A abelha, por questões biológicas, tem como produto o mel, onde é destinado ao sustento do enxame, bem como o seu armazenamento para épocas de escassez de flores.

O homem, com sua capacidade de observação aliado ao seu conhecimento, estudou os hábitos das abelhas e acabou descobrindo técnicas de controle da produção de mel e passou a explorá-las surgindo então o primeiro apicultor, caracterizado como um elemento que cria e cuida das abelhas de forma racional (WIESE, 1985).

Segundo Wiese (1985), a constituição de uma colônia em condições normais se caracteriza pela seguinte quantidade de indivíduos: 60.000 a 80.000 abelhas operárias, 1 rainha e de 0 a 400 zangões (machos).

O ciclo de vida das abelhas pode se distinguir em três fases, pois elas passam por uma metamorfose completa. O ovo, fecundado ou não; larva (branca, passando depois a cinza escuro); pré-pupa (a larva tece o casulo e a cela já está operculada); pupa ou ninfa, com a extremidade livre e depois inseto adulto (VIEIRA, 1989).

A tabela 2 mostra os respectivos indivíduos, contemplando o tempo (em dias) do seu ciclo total até atingir o inseto adulto.

Tabela 2 – Ciclo até atingir a fase adulta

	Ovo	Larva	Pupa	Total de Dias
Rainha	3	5 ½	7 ½	16
Operária	3	6	12	21
Zangão	3	6 ½	14 ½	24

Fonte: Adaptado de Wiese (1985).

Porém as abelhas, como todos os seres vivos, também morrem. E os apicultores sabem e tomam cuidado para manter em equilíbrio o seu enxame, sempre tomando cuidado para controlar a população da colmeia. As rainhas podem viver até 5 anos, mas em média sobrevivem apenas um ano. Operárias, entre 38 a 42 dias podendo chegar a 6 meses em clima com inverno prolongado e por final o zangão podendo chegar a 80 dias, quando a florada é boa ou há abundância de alimento na colmeia (WIESE, 1985).

## 2.4.2 Termorregulação

As abelhas são insetos sociais que evoluíram para regular a temperatura dentro do ninho através de uma série de mecanismos, que podem variar desde a seleção do local do ninho para otimizar a temperatura interna, ou até mesmo um comportamento simples como o deslocamento das crias para regiões dentro do próprio ninho, onde as temperaturas são mais favoráveis (DOMINGOS; GONÇALVES, 2014).

Os insetos têm seu metabolismo e atividade influenciados diretamente pela temperatura corpórea que, por sua vez, está quase que inteiramente na dependência da temperatura do ambiente. Temperaturas baixas normalmente inibem a atividade, enquanto as altas geralmente estimulam o animal (ALMEIDA, 2008).

Logo a temperatura é um elemento fundamental para se observar com bastante atenção. Independentemente da temperatura externa, a colmeia é mantida entre 34°C e 35°C (espécie *Apis mellifera mellifera*), que é a ideal para desenvolvimento das crias. Caso a temperatura esteja fora desta faixa, pode provocar aumento da mortalidade na colônia e as operárias que nascerem podem apresentar defeitos físicos nas asas ou em outras partes do corpo (CAMARGO et al, 2002).

Dentro de um ninho existem abelhas que coordenam as atividades de termorregulação de tal forma que a temperatura do ninho se mantenha estável. A quantidade de abelhas envolvidas neste trabalho de arrefecimento ou aquecimento irá aumentar ou diminuir de acordo com a temperatura atual do ninho (DOMINGOS; GONÇALVES, 2014). As abelhas contam com mecanismos para controlar a temperatura interna da colmeia, que parece ser tão complexo levando em consideração um animal tão pequeno.

Quando se deseja baixar a temperatura do interior da colônia, elas se distanciam dos favos e se aglomeram do lado de fora da caixa, algumas operárias ficam posicionadas na entrada do ninho, movimentando suas asas de forma a direcionar uma corrente de ar para o interior da colmeia. Essa corrente de ar, além de esfriar a colmeia, auxilia na evaporação da umidade do néctar, transformando-o em mel (CAMARGO et al, 2002).

Uma outra estratégia é o resfriamento evaporativo, onde as operárias buscam água e distribuem por todo o ninho para que a temperatura interna seja

reduzida através da evaporação da água. As abelhas podem formar barbas, fora da colônia, evitando liberar o calor metabólico dentro da colônia (DOMINGOS; GONÇALVES, 2014).

Quando se tem um cenário onde as abelhas buscam o oposto, ou seja, elevar a temperatura interna do ninho há outros mecanismos que são adotados. O principal deles se trata da mudança de comportamento, onde as abelhas se aglomeram (DOMINGOS; GONÇALVES, 2014). Caso o agrupamento ainda não seja suficiente, as operárias elevam sua taxa de metabolismo, provocando vibrações dos músculos torácicos, o que conseqüentemente acaba gerando calor. Também pode ocorrer uma troca de posição, onde as abelhas aglomeradas, localizadas no centro/núcleo, vão para as extremidades e vice-versa (CAMARGO et al., 2002).

Segundo estudos realizados por Almeida (2008), em temperaturas internas superiores a 41°C, as abelhas saem em massa da colônia, deixando as crias e alimento, o que caracteriza o abandono. Daí a tamanha importância em se aferir a temperatura de uma maneira contínua, visando manter a saúde da população de uma colmeia. Essa possibilidade permite também averiguar questões como monitoramento se os mecanismos do controle da temperatura interna estão se mostrando eficientes e eficazes.



### 3 TECNOLOGIAS PARA O DESENVOLVIMENTO DO PROTÓTIPO

Diversos tipos de tecnologias estão presente no dia a dia de milhares de pessoas, demonstrando a sua popularização e trazendo inúmeras vantagens para as mais diversas áreas e funções. Neste capítulo serão abordados diferentes tipos delas afim de caracterizar possibilidades para constituir o projeto proposto.

#### 3.1 INTERNET DAS COISAS

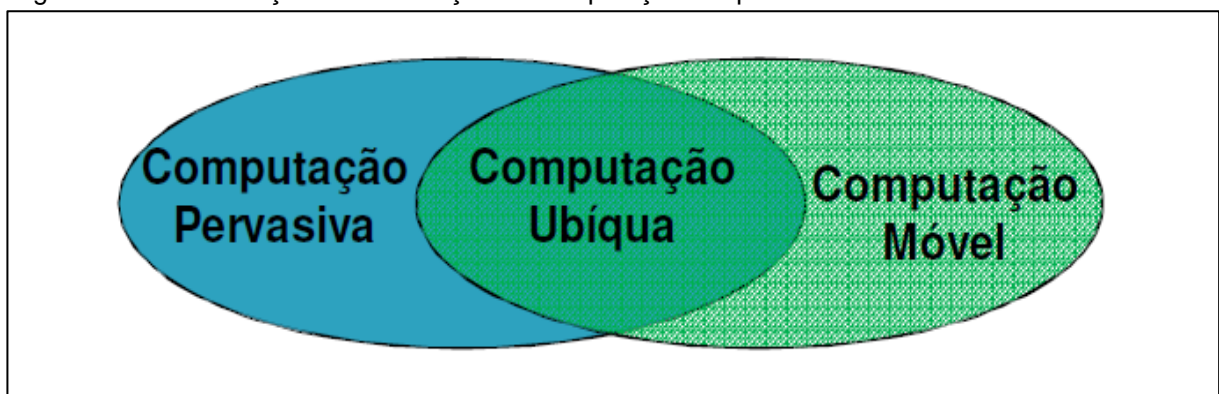
O termo Internet das Coisas ou *Internet of Things* (IoT) foi citado em 1999 por Kevin Ashton, cofundador do *Auto-ID Center no Massachusetts Institute of Technology* (MIT) que sustenta ter sido o primeiro a ter utilizado este termo (ASHTON, 2009, tradução nossa). Trata-se de um novo paradigma que surgiu e está rapidamente ganhando terreno no cenário das telecomunicações. A ideia básica desse conceito é a presença generalizada em torno de nós de uma variedade de coisas ou objetos, através de uma conexão com a Internet - como *Radio-Frequency IDentification* (RFID), sensores, celulares, etiquetas, lâmpadas, entre outros - para alcançar objetivos comuns (ATZORI et al., 2010, tradução nossa).

Essa tecnologia está inserida em uma definição mais ampla que se trata da computação ubíqua. Marc Weiser (1991), idealizador da computação ubíqua, profetizou há mais de uma década que no futuro os computadores estariam presentes nos mais comuns objetos do dia-a-dia: etiquetas de roupas, xícaras de café, interruptores de luz, canetas, entre outros, de forma invisível para o usuário. Esse ambiente oculto para o usuário vai ao encontro do conceito de computação pervasiva. Nesta concepção, o computador tem a capacidade de obter informação do ambiente no qual ele está inserido e utilizá-la dinamicamente para gerenciar modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação para melhor atender as necessidades do dispositivo ou usuário. (ARAUJO, 2003).

Entende-se como computação ubíqua a união de dois conceitos bases: a computação móvel juntamente com a pervasiva. A soma destas duas tecnologias contemporâneas resultou no terceiro paradigma, o qual foge da característica de Computadores Pessoais, do inglês *Personal Computer* (PC), e estações de trabalhos.

A computação móvel se trata de uma tecnologia extremamente popular no mundo. Ela tem como característica a capacidade de mover fisicamente serviços computacionais com o usuário. O computador torna-se um dispositivo onipresente que aumenta a capacidade de um usuário utilizar seus serviços, independentemente de sua localização. Aliada com a capacidade de acesso, a computação móvel tem transformado a computação em uma atividade que pode ser levada para qualquer lugar (ARAÚJO, 2003). Figura 3 demonstra de forma ilustrada a intersecção.

Figura 3 – Demonstração da intersecção da computação ubíqua.



Fonte: Cirilo (2008)

Internet das Coisas agrega a capacidade de integrar objetos, ou coisas como o nome sugere, com intuito de estabelecer uma comunicação entre eles, desde que estejam inseridos a uma rede em comum e com acesso à Internet possibilitando assim uma interação de forma fluída. Havendo esta possibilidade, os objetos podem coletar e posteriormente trocar informações de tal ambiente, seja ela de qualquer natureza. Com estes dados e autonomia, os sensores podem acionar processos, criar serviços ou até então disparar ações independentes de intervenção humana.

Segundo Santucci (2010) a atual Internet dos PCs irá avançar para uma Internet das Coisas em que cinquenta a cem bilhões de dispositivos serão conectados à Internet até 2020. Visando estes números expoentes, mostra-se uma vantagem estar desenvolvendo o protótipo dentro do cenário de IoT, uma vez que tal tecnologia vai de encontro com um ponto muito importante dentro deste projeto de pesquisa, que se trata de desenvolver um sistema de monitoramento em uma colmeia utilizando as tecnologias necessárias. O protótipo visa aferir condições da natureza de uma colmeia como a temperatura, o peso, os ruídos internos e a umidade relativa do ar e IoT tem

como essência fazer essa comunicação de diferentes objetos vinculados a uma rede em comum.

Esta integração não irá se ater somente à Internet das Coisas, mas também embarca outras tecnologias para que o sistema proposto entregue as aferições de uma maneira simples de leitura, como um aplicativo.

### 3.2 PLATAFORMAS MÓVEIS

A mais recente era de aparelhos móveis agregam inúmeras possibilidades como *smartphones*, *tablets* e iPods. Essas opções contemplam uma infinidade de ferramentas embutidas, como por exemplo opções para personalização, lojas virtuais (*App Store* da *Apple*, *Play Store* da Google). Essas lojas dispõem das mais variadas categorias de aplicativos sejam eles gratuitos ou pagos, que vão de jogos, gerenciadores financeiros, redes sociais e muito mais, sempre tentando contentar todos os tipos de consumidores (PILAR, 2013).

A característica de um dispositivo móvel está atrelada a mobilidade, que é a capacidade de chegar aos lugares necessários na vida urbana, como trabalho, escola, comércio, além de realizar múltiplas e diferentes tarefas, bem como conectar-se e trocar informações com outros dispositivos, pessoas e sistemas (PILAR, 2013).

Para ser considerado móvel, um dispositivo deve possuir quatro características essenciais (LEE, 2005), sendo elas:

- a) portabilidade: capacidade do dispositivo de se deslocar ou ser deslocado facilmente, onde em geral devem poder ser carregados na mão. Este fator abrange tamanho e peso, incluindo também os seus acessórios;
- b) usabilidade: é a capacidade do dispositivo de ser utilizado por diferentes pessoas e ambientes.
- c) funcionalidade: são as tarefas que o dispositivo pode efetuar através de aplicações móveis que executa
- d) conectividade: dispositivos móveis têm como principal característica a conectividade, pois não são feitos para operarem sozinhos. O seu objetivo é poder conectar pessoas e trocar informações entre diferentes sistemas. A conectividade pode ser dividida em três formas: estar

sempre conectado, conexões intermitentes e operações que não necessitam de conexão. (LEE, 2005).

As plataformas móveis formam um conjunto de tecnologias que agregam vários recursos. Com isso os *smartphones* passaram a utilizar plataformas para o funcionamento. Essas plataformas possuem a responsabilidade em gerenciar todas as funções de *hardware* de um dispositivo móvel, como GPS, câmera, agenda entre outros recursos disponíveis (ALLEN; GRAUPERA; LUNDRIGAN, 2012). Logo após o surgimento dele foi possível notar a diferença entre as tecnologias, uma vez que o *smartphone* mudou totalmente a percepção que se tinha a respeito da telefonia móvel, com as funcionalidades como, *bluetooth*, redes 4g, reconhecimento facial, reconhecimento biométrico entre outros (TELECO, 2014).

Naturalmente os aparelhos mais fáceis de utilizar e com diversas funcionalidades e com maior custo-benefício tem maior procura por parte dos consumidores. Os desenvolvedores, por outro lado, procuram por *smartphones* ou *tablets*, onde a plataforma tem maior reconhecimento de mercado e que tenha uma boa documentação, baixo custo de desenvolvimento entre outros (ALLEN; GRAUPERA; LUNDRIGAN, 2012).

Cada plataforma possui suas peculiaridades com seus pontos fortes e suas desvantagens. Possuem linguagens de desenvolvimento própria, permitindo criar aplicativos para posteriormente serem disponibilizados em suas lojas virtuais.

### 3.2.1 Plataforma Android

O Android é uma plataforma móvel de código aberto, seu *kernel* é baseado em Linux. Originalmente seu desenvolvimento foi feito por uma *startup* com o mesmo nome: Android. Em 2005 a Google acabou comprando e adquirindo os direitos. Logo em 2007 foi anunciado que no ano seguinte o sistema passou a ser disponibilizado através do *smartphone* HTC Dream na sua versão 1.0. Em seguida foi lançado para o Motorola Droid e o HTC Nexus One. Em 2011 passou a ser disponibilizado também para *tablets*, iniciando através do Motorola Xoom.

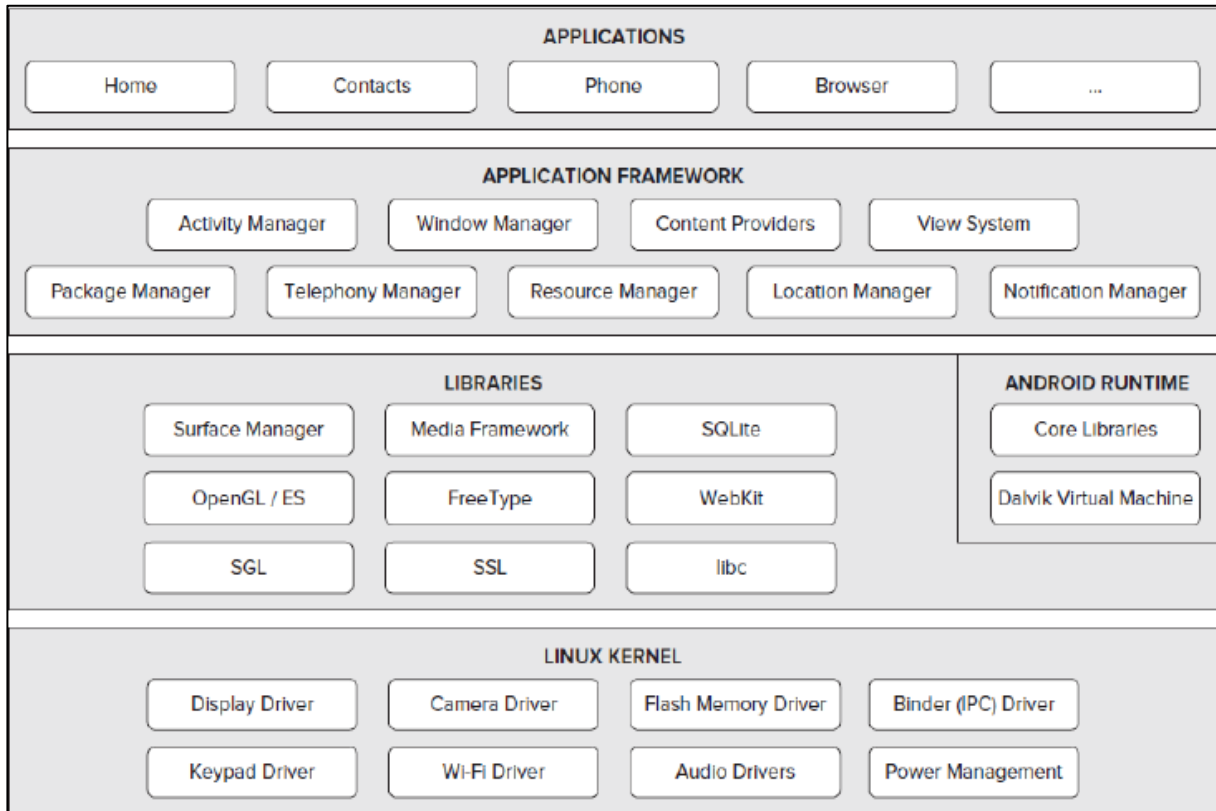
Curiosamente, as versões do sistema seguem uma ordem alfabética e a nomenclatura de doces, iniciando com o CupCake, Donut e Éclair em 2009, seguido pelo Froyo e Gingerbread em 2010, pelo Honeycomb e Ice Cream em 2011.

As principais vantagens embarcadas do Android são (GOOGLE, 2013):

- a) utilização de *widgets*: pequenos ícones que ficam na tela inicial, onde é possível escolher entre seus aplicativos favoritos, tendo acesso a atualizações sem sair da tela principal;
- b) tela de notificações: onde é possível visualizar em um só lugar as notificações de novas ligações, mensagens e e-mails;
- c) multitarefas: é possível executar diversos aplicativos ao mesmo tempo e alternar a visualização entre eles;
- d) escrita por voz: possibilidade de escrever e-mails, SMS e qualquer tipo de texto, através da voz;
- e) compartilhamento de imagens: possibilidade de explorar, editar e compartilhar rapidamente nas redes sociais as fotos tiradas direto do celular;
- f) compartilhar com o toque: compartilhamento de fotos e vídeos com outros celulares com a mesma tecnologia, apenas encostando-os. Também é possível parear dispositivos via *bluetooth* apenas encostando os aparelhos;
- g) múltiplos usuários: nos tablets, é possível criar múltiplos usuários, cada um com sua tela inicial, aplicativos e jogos.

A arquitetura do sistema operacional Android pode ser dividida em cinco camadas (LEE, 2012), conforme exemplificado na figura 4.

Figura 4 – Camadas Android



Fonte: Arquitetura Android (LEE, 2012)

As camadas representadas na figura 4 possuem as seguintes características (LEE, 2012, tradução nossa):

- aplicativo: nesta camada encontra-se todos os aplicativos já instalados no dispositivo e aqueles que o usuário faz o *download* por meio da loja virtual;
- Framework* de aplicação: camada na qual são disponibilizadas as funcionalidades para os desenvolvedores utilizarem nos aplicativos;
- Bibliotecas: seção onde contém o código fonte das funcionalidades da tecnologia como por exemplo suporte ao banco de dados SQLite.
- Runtime* Android: é a mesma camada da biblioteca, tem como função possibilitar um conjunto de bibliotecas que permitem ao desenvolvedor programar utilizando a linguagem de programação Java e também a linguagem mais recentemente denominada Kotlin;

- e) Núcleo Linux: o núcleo na qual é embasado o Android. Essa camada abriga todos os *drivers* de baixo nível para os diversos componentes de um dispositivo Android.

Para o desenvolvimento de uma aplicação Android são necessárias algumas ferramentas. Essas ferramentas são gratuitas, o que é um grande ponto a ser observado quando comparado com outras soluções *mobile*. O Android SDK é um *software* utilizado para desenvolver aplicativos destinados a dispositivos que tenham o sistema Android embarcado. Têm um emulador para simular celulares, ferramentas utilitárias e uma API completa para a linguagem JAVA e Kotlin, repleta de classes necessárias para o desenvolvimento de uma aplicação (LECHETA, 2015).

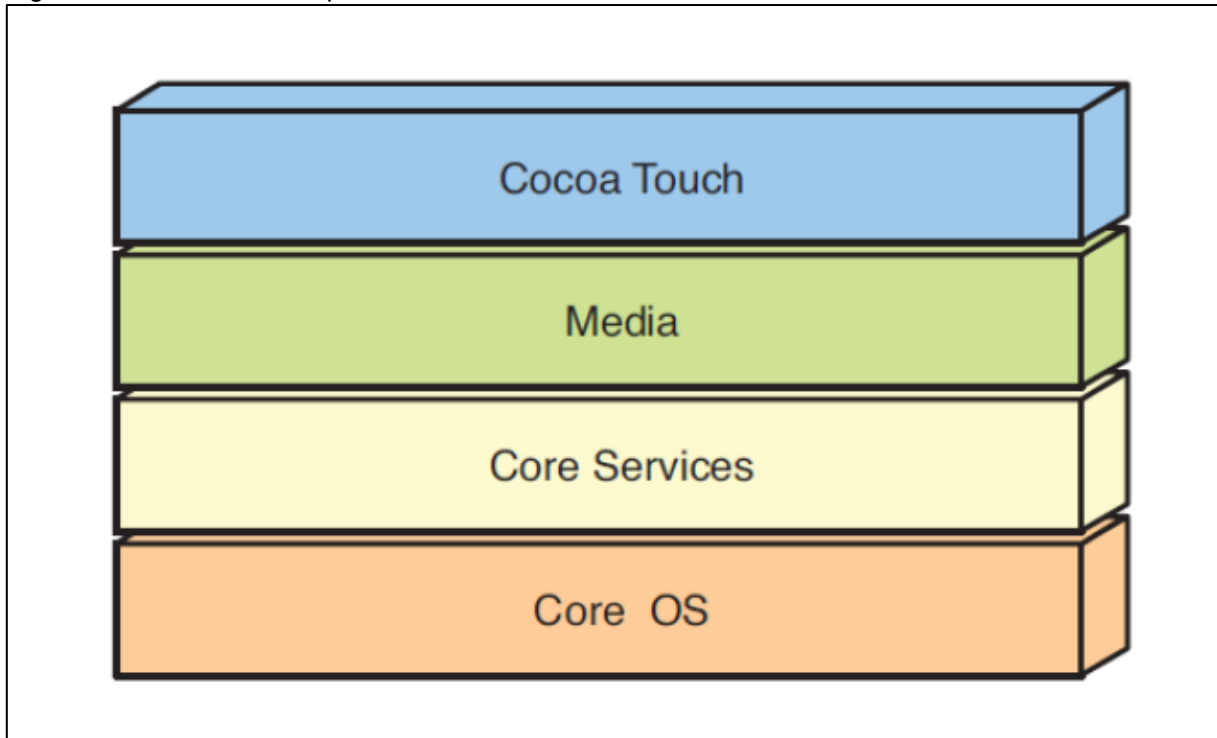
### 3.2.2 Plataforma iOS

É uma plataforma proprietária da *Apple* que está presente em todos os seus dispositivos móveis, como *tablet* e *smartphones*. O iOS foi lançado em 2007 juntamente com o iPhone e era denominado de iPhone OS. Somente em 2010 seu nome foi alterado para iOS, quando outros dispositivos passaram a utilizar o sistema operacional, como iPod's e iPad's (PILAR, 2013).

Logo em seu primeiro ano o iPhone impressionou com sua tela *multitouch* com uma ótima sensibilidade e também pela disposição dos recursos e sua integração com o acelerômetro, GPS, serviços e entre outros. Mas o que realmente alavancou o desenvolvimento para esse sistema foi a loja *App Store*, um grande feito de Steve Jobs, possibilitando que aplicativos fossem distribuídos e instalados facilmente em qualquer aparelho e ainda com uma proposta de remuneração aos desenvolvedores, o que foi uma grande revolução para o mercado. (LANCHETA, 2014).

A arquitetura do sistema operacional iOS é dividida em quatro camadas (RAY, 2011), conforme figura 5.

Figura 5 – Camadas da arquitetura iOS



Fonte: Ray (2011)

As camadas da arquitetura podem ser caracterizadas com suas respectivas funcionalidades (ROCHA; FINZI, 2014):

- a) *cocoa touch*: camada formada pelos *frameworks* para a construção das aplicações, define a infraestrutura necessária e diversos serviços de alto nível do sistema;
- b) *media services*: contém as tecnologias multimídias, como gráficos, vídeos e áudio. Essas ferramentas tornam mais fácil o desenvolvimento de aplicações multimídia;
- c) *core services*: nesta camada encontram-se as funcionalidades de acesso aos serviços fundamentais do sistema operacional. As principais tecnologias disponíveis nessa seção são: *grand central dispatch*, *in-app purchase*, *SQLite* e *XML support*;



- d) *core OS*: é a camada mais próxima ao *hardware* do dispositivo, característica de mais baixo nível do sistema operacional, como *threads*, cálculos matemáticos, acessórios de *hardware* e criptografias.

### 3.3 APLICAÇÕES MÓVEIS

Hoje existem muitas plataformas para desenvolvimento móvel, como Android, iOS, Windows Phone, entre outros. Cada uma possui sua peculiaridade, como diferentes formas e linguagens para desenvolvimento, tendo como por exemplo o Android utiliza Java ou Kotlin como linguagem de programação e o iOS utiliza *Objective C* ou atualmente *Swift* (PREZZOTO; BONIATI, 2014).

Diante dessa variedade de possibilidades para se construir uma aplicação, os desenvolvedores e as empresas buscam cada vez mais meios para reduzir manutenções e custos. Na grande maioria não se tem um aproveitamento de código fonte obrigando os desenvolvedores a reescreverem o algoritmo das aplicações para que a mesma esteja disponível para mais de uma plataforma (DEVMEDIA, 2018). A tabela 3 compara os sistemas embarcados nos dispositivos móveis com suas respectivas linguagens, Ambiente de Desenvolvimento Integrado, do inglês *Integrated Development Environment* (IDE) e Sistema Operacional (SO) de desenvolvimento.

Tabela 3 – Sistemas de dispositivos móveis e suas respectivas características

Sistema Operacional	Linguagem de desenvolvimento	IDE	Sistema Operacional de desenvolvimento
iOS	Objective C/ <i>Swift</i>	XCode	MacOS
Android	Java	Eclipse/Android Studio	Windows, Linux, MacOS
Windows Phone	C#	Visual Studio	Windows

Fonte: Adaptado de DEVMEDIA (2018).

Atualmente existem três meios para se desenvolver um aplicativo móvel. São elas: nativas, híbridas e *webapps*. Na tabela 4 faz-se um comparativo entre essas alternativas de desenvolvimento salientando as vantagens e desvantagens das mesmas.

Tabela 4 – Comparativo das possibilidades de desenvolvimento

Critério	Nativa	Híbrida	Webapps
Multiplataforma	Não	Sim	Sim
Custo de Desenvolvimento	Alto	Baixo	Baixo/Médio
Desempenho	Alto	Baixo/Médio	Médio
Acesso ao dispositivo	Total	Limitado	Total
Utilização off-line	Sim	Sim	Sim
Loja de Aplicativo	Sim	Não	Sim
Facilidade de Manutenção	Baixa	Alta	Alta

Fonte: Adaptado de Montan; Santos (2017).

Os subcapítulos desta seção irão discorrer mais a respeito dessas tecnologias afim de salientar pontos, bem como servirão para tomada de decisão do vigente projeto de pesquisa que será utilizado na construção do protótipo do monitoramento das colmeias.

### 3.3.1 Aplicações Nativas

As aplicações nativas têm como propósito desenvolver uma solução para uma plataforma específica, utilizando ferramentas disponibilizadas pela detentora da plataforma como por exemplo a linguagem de desenvolvimento, ambiente e emulador. Aplicativos desenvolvidos na linguagem de programação nativa do sistema tem um melhor desempenho, embora as outras tecnologias procurem evoluir neste quesito. Um ambiente nativo já estabelecido auxilia muito no processo, que se for necessário desenvolver para sistemas distintos acaba se tornando irrelevante, pois em cada sistema é necessário desenvolver uma aplicação específicas diferindo assim das outras formas de desenvolvimento (PREZOTTO; BONIATI, 2014).

Embora o desempenho e a interface gráfica em aplicações nativas tendem a serem melhores, desenvolver uma aplicação para cada plataforma acarreta em maiores custos, esforço e tempo (MONTAN; SANTOS, 2017). Esse desenvolvimento acaba tendo um custo benefício abalado, uma vez que é necessário um conhecimento concreto de cada linguagem de programação específica na qual a plataforma

contempla e levando em consideração a quantidade de linguagens hoje no mercado, se torna difícil ter um discernimento diante de tantas possibilidades.

### 3.3.2 Aplicações Híbridas

Aplicações híbridas são uma categoria especial de aplicativos da *web* que ampliam o ambiente de aplicativos por meio de APIs de plataforma nativas disponíveis em um determinado dispositivo (GOK; KHANNA, 2013, tradução nossa). Possuem como finalidade funcionar nos mais diferentes dispositivos, sendo que para diferentes plataformas, será utilizado o mesmo código fonte. Ficam instaladas no dispositivo e podem funcionar de forma independente da conexão com a Internet.

Partindo do princípio de utilizar a mesma aplicação em diferentes plataformas, normalmente as aplicações híbridas são desenvolvidas em linguagens *web* que são interpretadas pelo *browser* nativo do sistema (PREZOTTO; BONIATI, 2014).

Dentro da aplicação híbrida há três abordagens para o seu desenvolvimento, são elas: tempo de execução, tradutor de código e Web-para-Nativo.

Na opção tempo de execução o aplicativo é desenvolvido em uma linguagem *script* (*JavaScript*, Lua ou Ruby). A ferramenta para o desenvolvimento cria um pacote de instalação do aplicativo com o interpretador da linguagem utilizada e o código *script*. Em tempo de execução, o interpretador traduz o código *script*/fonte para código de máquina (nativo) para então ser executado (MONTAN; SANTOS, 2017).

Os tradutores de código se mostram semelhante ao tempo de execução, porém com duas diferenças. A primeira está atrelada a linguagem de desenvolvimento que não se limita a linguagens *script*. A segunda está na parte da compilação: o código fonte é compilado para *byte-code* ou código de máquina (nativo). O *byte-code* é universal existindo para as principais plataformas (MONTANAN; SANTOS, 2017).

A abordagem Web-para-Nativo (Web-To-Native Wrapper) alia as vantagens das *webapps* juntamente com aplicações nativas. São utilizadas linguagens de desenvolvimento WEB (HTML, CSS e *JavaScript*), porém os aplicativos não são executados por um navegador. A aplicação é executada por um container nativo (WebView no Android e UIWebView no iOS) e o acesso ao hardware e recursos

do dispositivo é feito por bibliotecas. As principais desvantagens estão ligadas a interface ser distinta da interface nativa da plataforma, já conhecida pelo usuário, e o desempenho, prejudicado pela execução no container (MONTAN; SANTOS, 2017).

### 3.3.2.1 React Native

React Native<sup>1</sup> é um framework *JavaScript* para desenvolver aplicações móveis em diferentes sistemas operacionais, no caso iOS e Android. É baseada no React, uma biblioteca web desenvolvida pelos engenheiros do Facebook. Em outras palavras, no desenvolvimento web, é possível escrever aplicativos que parecem realmente "nativos", tudo a partir da flexibilidade e poder do *JavaScript*. (Eisenman, 2015, tradução nossa). Ele faz uso do Node.js para efetuar o *build*<sup>2</sup> (construção) de código *JavaScript*, uma tecnologia recente que vem ganhando mercado.

O conceito principal é “aprenda uma vez, escreva em qualquer lugar”. Isso significa um desenvolvedor pode criar um aplicativo web por meio do ReactJS e também pode criar aplicativos móveis através do React Native sem experiência anterior em desenvolvimento nativo devido a semelhança entre estas duas tecnologias (Vidhall, Hansson, 2016, tradução nossa).

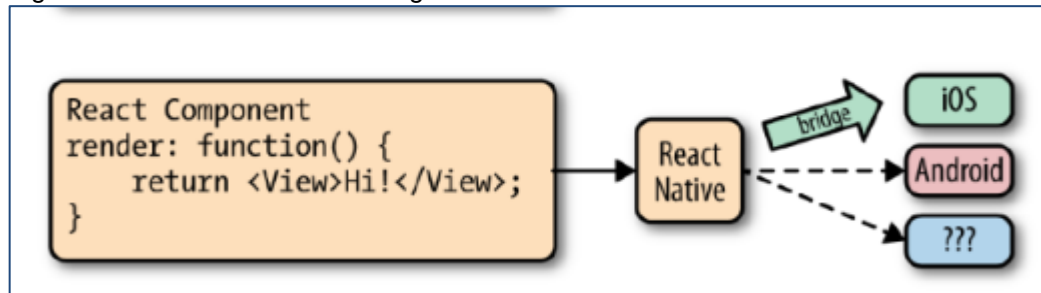
Um ponto muito importante desta tecnologia está associada a sua *bridge*, que seria a ponte entre o javascript e as linguagens nativas (Swift e Java/Kotlin). Os componentes deste *framework* envolvem o código nativo existente e interagem com as APIs nativas por meio do paradigma declarativo da interface do usuário e do *JavaScript* (React Native, 2019). A figura 6 exemplifica como a *bridge* atua no *framework*.

---

<sup>1</sup> <https://facebook.github.io/react-native/>

<sup>2</sup> Versão compilada do código

Figura 6 – Funcionamento da bridge no React Native



Fonte: Eisenman (2015)

Esta característica de ponta acaba diferenciando o React Native de outras opções de desenvolvimento de aplicativos híbridos, que geralmente acabam renderizando visualizações na web. O código é traduzido para se adequar à plataforma; portanto, uma tela pode se adequar ao modo de exibição específico do iOS. Devido à camada de abstração, como demonstrado na figura 6, o React Native também pode ter como alvo outras plataformas – se faz necessário a existência de uma *bridge*.

O fato de o React Native renderizar seus elementos usando a API nativa da plataforma em questão leva a se destacar da maioria das tecnologias existentes no desenvolvimento de aplicativos híbridos, como Cordova ou Ionic. Os métodos existentes para escrever aplicativos móveis usando combinações de *JavaScript*, HTML e CSS normalmente são renderizados usando a visualização na web. Embora essa abordagem possa funcionar, ela também apresenta desvantagens, especialmente em torno do desempenho. Além disso, essas tecnologias geralmente não têm acesso ao conjunto de ferramentas disponibilizadas pelo sistema operacional. (Eisenman, 2015, tradução nossa).

### 3.3.3 Aplicações WebApps

São aplicações *web* que funcionam no *browser* do dispositivo, assim não é necessário se preocupar com variações nas linguagens de programação, pois utiliza-se padrões *web* como HTML, CSS e *JavaScript*. Embora empregado a mesma linguagem para todos os sistemas operacionais, o comportamento em cada uma delas depende da maneira de como o navegador processa a própria aplicação (PREZOTTO; BONIATI, 2014).

Diferentemente das aplicações nativas, não há a necessidade de uma instalação, já que as mesmas são acessadas através do *browser*. Um dos grandes fatores que influencia no funcionamento dessas aplicações, é a conexão estável com a Internet, uma vez que ela é a principal responsável por garantir velocidade e disponibilidade da aplicação (WILKEN, 2015, tradução nossa).

*Webapps* apresentam algumas desvantagens se comparadas com aplicações nativas. O acesso aos recursos do dispositivo é limitado a aplicação não é instalada no dispositivo e geralmente só funciona com o dispositivo conectado à Internet (MONTAN; SANTOS, 2017). Outra desvantagem referente a *webapps* é referente ao seu *layout*, que necessita ser reestruturado diversas vezes já que os *smartphones* e *tablets* possuem tamanhos distintos de tela, sendo assim o desenvolvimento é muito focado nestes pequenos detalhes, porém, atualmente com *frameworks* de UI responsivos, isto deixou de ser um dos grandes problemas (WILKEN, 2015, tradução nossa).

## 4 MICROCONTROLADOR E SENSORES

Os microcontroladores possuem características de um sistema computacional completo em um único *chip* e são encontrados em uma grande variedade de dispositivos. Possuem internamente processador, memória para armazenar dados do tipo RAM (*Random Access Memory*), memória para armazenamento de programa do tipo EEPROM (*Electrically Erasable Programmable Read-Only Memory*), capacidade de entrada e saída e de comunicação, timer gerador de sinais de interrupção, conversor A/D (Analógico Digital), entre outras características específicas de cada fabricante (ORDONEZ et al., 2003).

Um sensor se trata de um dispositivo que detecta e sinaliza as mudanças de condições de um ambiente. Por exemplo, um sensor pode sinalizar a presença ou ausência de um objeto ou material. Um sensor analógico pode aferir grandezas como distância, tamanho e cor, entre outras (ORDONEZ; PENTEADO; SILVA, 2006).

Tendo como estas duas caracterizações o projeto de pesquisa visa empregar ambos no desenvolvimento do protótipo. Exemplificando as opções disponíveis de sensores e da sua utilização, unindo sensores e um microcontrolador afim de aferir as condições de uma colmeia.

### 4.1 ARDUINO

O Arduino teve seu início no *Interaction Design Institute* (Instituto de Design Interativo, em tradução livre), em 2005. Massimo Banzi, professor, procurava um meio que fosse barato e fácil para que os estudantes de design pudessem trabalhar com tecnologia. Juntamente com David Cuartielles, que buscava uma solução semelhante, se propuseram a criar uma alternativa, uma vez que as que existiam no mercado eram caras e relativamente complicadas (EVANS; NOBLE; HOCHENBAUM, 2013).

As principais exigências eram que fosse barato - o preço desejado não deveria ultrapassar o que um estudante gastaria se saísse para comer uma pizza - e que fosse uma plataforma que qualquer pessoa pudesse utilizar. Cuartielles desenhou a placa e um aluno, David Mellis, desenvolveu a programação. A nova placa foi chamada de Arduino em referência a um bar local frequentado pelo corpo docente e alunos da instituição (EVANS; NOBLE; HOCHENBAUM, 2013).

É uma plataforma para prototipagem de objetos interativos usando eletrônica. É composto por um conjunto de *hardware* e *software*: uma placa de circuito que pode ser adquirida por um preço acessível e contém um ambiente de desenvolvimento de código aberto, com bibliotecas, onde se pode controlar a placa. (MELLIS et al., 2007, tradução nossa).

A plataforma tem uma conexão USB na qual permite a troca de dados com computadores e com esta comunicação, ela permite o controle de diversos dispositivos externos como sensores, relés e motores (MONK, 2014).

Com a possibilidades de diferentes fontes de alimentação, o que se torna um ponto positivo para a sua utilização, a placa pode ser alimentada por meio do cabo USB, ou então, utilizar energia elétrica externa por meio de um adaptador de tensão contínua (MONK, 2014).

Em seu site oficial (ARDUINO, 2018) o Arduino se define como “uma plataforma eletrônica de código aberto baseada em hardware e software de fácil utilização. É destinado para que qualquer pessoa possa criar projetos interativos”.

A figura 7 mostra o Arduino Uno Rev3 que é uma das várias possibilidades que se tem para aplicar dentro do projeto.

Figura 7 – Arduino UNO Rev 3











Fonte: Site oficial do Arduino (2018)



Além do Arduino UNO existem outros para estar empregando no protótipo a ser desenvolvido. Na figura 8 demonstra outros modelos bem como suas especificações técnicas, fazendo um breve comparativo.

Figura 8 – Comparativo entre as opções de Arduino

	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v

Fonte: Coelho (2015)

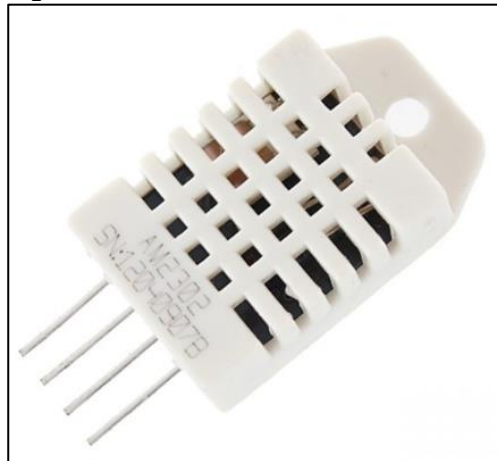
Com estas características, o Arduino é uma ótima opção, também se considerando o seu custo benefício e sua versatilidade. A escolha dessa plataforma para o protótipo a ser desenvolvido se deu devido a opção de incorporar diversos sensores ao seu *hardware*, os quais poderão ser gerenciados por meio de uma interface de programação de fácil compreensão, além de contar com uma comunidade entusiasta e a possibilidade de encontrar diversos códigos abertos e prontos para utilização.

## 4.2 SENSORES DE TEMPERATURA E UMIDADE

Sensores de temperatura e umidade são dispositivos que tem como funcionalidades aferir grandezas físicas ligadas à temperatura, geralmente a variação é dada em graus Celsius, e umidade relativa do ar, que nada mais é do que a proporção de água existente no ar em determinado ambiente em que estão acoplados. Existe uma diversidade de sensores com essas características embutidas, podendo citar rapidamente o modelo AM2302 DHT22 e o HDC1080.

O modelo AM2302 DHT22 é um sensor de temperatura e umidade que permite fazer leituras de temperatura de uma faixa que compreende entre -40 a +80 graus Celsius e umidade entre 0 chegando a 100% e é de fácil utilização diante do Arduino, pois possui apenas um pino com saída digital (AOSONG, 2019, tradução nossa). A figura 9 ilustra o sensor mencionado anteriormente, o AM2302 DHT22.

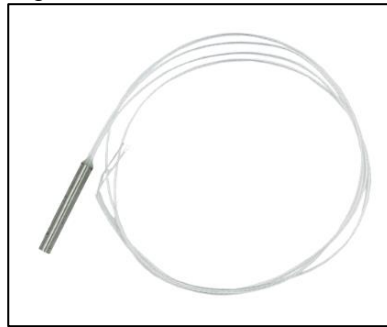
Figura 9 – Sensor AM2302 DHT22



Fonte: Insituto NCB(2018)

O sensor PT 100 (figura 10) tem a característica de medir apenas a temperatura, diferentemente do AM2302 DHT22. O modelo PT 100 dispõe de termorresistências de platina que são sensores de temperatura cujo princípio de funcionamento baseia-se na alteração da resistência elétrica com a variação da temperatura. Embora tenha uma resistência elétrica baixa, o PT 100 é capaz de medir temperaturas entre -20 a 450 graus Celsius, além de ser à prova d'água.

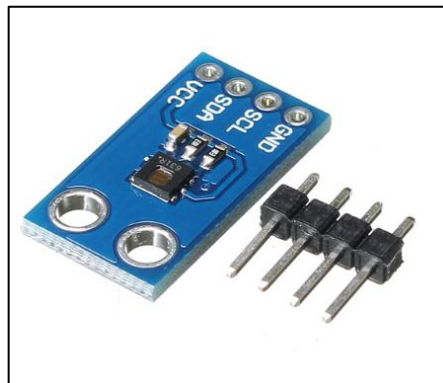
Figura 10 – Sensor PT 100



Fonte: Electron (2018)

O modelo HDC1080 é um sensor de alta precisão capaz de medir umidade bem como temperatura e pode ser encontrado em equipamentos meteorológicos e também em médico hospitalares. Ele possui catorze *bits* de resolução e precisão de umidade de 2% e de temperatura de 0.2 °C, fornecendo informações consistentes confiáveis a um baixo custo de energia (DUARTE, 2017). A figura 11 ilustra o sensor de alta precisão HDC1080.

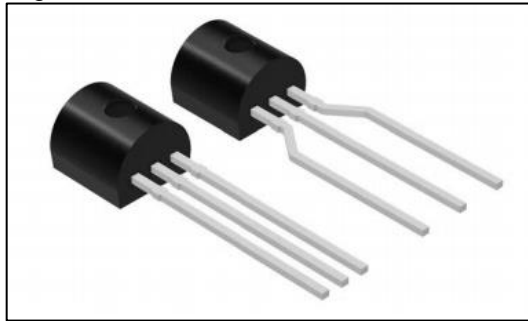
Figura 11 – Sensor HDC1080



Fonte: Texas Instruments (2018)

O sensor de temperatura LM35 é um dos mais comuns e compatibilidade de acoplamento ao Arduino. Esse dispositivo é capaz de medir uma faixa de temperatura entre 0°C a 100°C, atua com uma calibração em graus Celsius e tem como tensão uma variação de 3V a 30V, conforme a figura 12.

Figura 12 – Sensor LM35



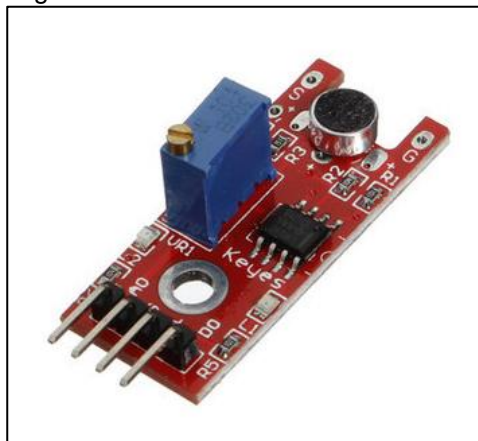
Fonte: Texas Instruments (2017).

## 4.2 SENSORES DE SOM

Sensores de som tem como finalidade a detecção de ruídos, medindo a intensidade sonora do ambiente ao redor, variando o estado de sua saída digital caso detectado um sinal sonoro. Pode-se citar um módulo com esta finalidade o KY-038.

Esse sensor opera com uma tensão entre 4V e 6V, tem um led indicador para sua tensão e para sua saída digital, permite o ajuste de sua sensibilidade por meio de um potenciômetro e é de fácil instalação (MASTERWALKER, 2018). A figura 13 ilustra o módulo KY-038.

Figura 13 – Sensor de ruído KY-038



Fonte: Datasheetcafe (2018)

### 4.3 SENSORES DE PESO

Os sensores de peso têm como funcionalidade medir grandezas da área da massa de um determinado corpo, utilizando-se da gravidade para realização do seu cálculo. Pode-se citar o sensor de carga *Strain Gauge*. Seu funcionamento se dá devido às mudanças de resistências elétricas proporcional a uma determinada pressão ou força aplicada sobre ele. Tem a capacidade de pesar até cinquenta quilogramas contando com uma margem de erro de 0,2% e é alimentado por uma tensão entre 5V e 10V. A figura 14 exemplifica o sensor em questão.

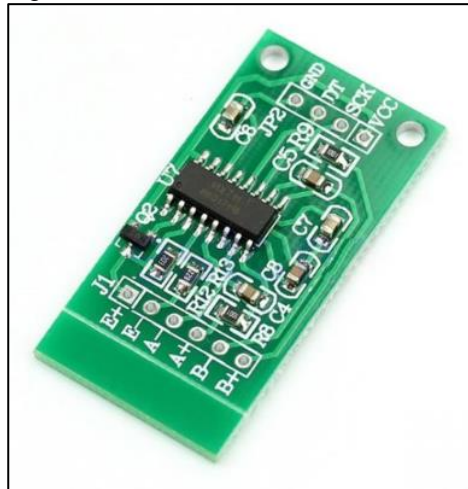
Figura 14 – Célula de carga



Fonte: Vidal (2018)

Para que sua utilização seja eficiente, precisa-se de um módulo conversor para amplificar o sinal do dispositivo da célula de carga, interligando a célula e o microcontrolador, esse módulo é o HX711 e é mostrado na figura 15.

Figura 15 – Módulo HX711



Fonte: Vidal (2018)

#### 4.4 CONEXÕES COM DISPOSITIVOS EXTERNOS

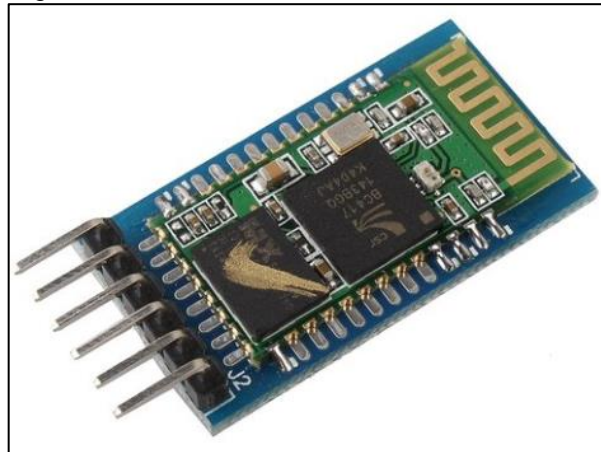
Para que o protótipo se mostre funcional, faz-se necessário estabelecer uma comunicação dos sensores com dispositivos externos para a visualização dos dados capturados. Nas subseções a seguir serão abordadas tecnologias que tem essa característica de manter uma comunicação entre dois dispositivos.

##### 4.4.1 Módulo Bluetooth

O *Bluetooth* é uma tecnologia de comunicação sem fio desenvolvida pela empresa de telecomunicações Ericsson em 1994. Permitindo a troca de dados e arquivos entre celulares, computadores, scanners, fones de ouvidos e uma outra série de dispositivos garantindo velocidade e segurança. O sistema utiliza uma frequência de rádio de onda curta, ou seja 2.4Ghz para estabelecer uma comunicação entre os dispositivos pareados. Seu alcance é curto e só permite troca de dados entre dispositivos próximos e seu consumo de energia é relativamente baixo (CÂMARA, 2012).

Pode-se citar um módulo *bluetooth* HC-05 que acoplado ao Arduino possibilita esta conexão. O HC-05 tem como especificação o modo escravo (*slave*), bem como o modo mestre (*master*), além de ser de fácil configuração. Possui um alcance de até dez metros e sua alimentação pode variar entre 3.3 V e 5 V. A figura 16 ilustra o sensor *bluetooth*.

Figura 16 – Módulo HC-05



Fonte: Oddwires (2019).

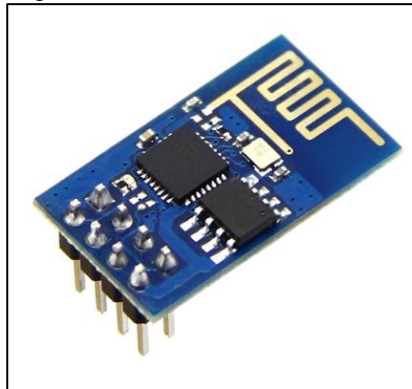
#### 4.4.2 Módulo Wifi

Apesar de o termo *Wi-Fi* ser uma marca registrada pela Wi-Fi Alliance, a expressão hoje se tornou um sinônimo para a tecnologia IEEE 802.11, que permite a conexão entre diversos dispositivos sem fio. Seu funcionamento se dá por meio de ondas de rádio que são transmitidas através de um adaptador chamado “roteador”. Esse roteador recebe os sinais, decodifica e os emite a partir de uma antena. Para que o computador ou dispositivo tenha acesso a esses sinais é preciso que ele esteja inserido em um determinado raio de ação, conhecido como *hotspot* (LANDIM, 2012).

Existem módulos que acoplados ao Arduino permitem a possibilidade do uso desta tecnologia, mas por outro lado tem placas que já vem com esta função de fábrica. Pode-se citar o módulo ESP8266 que é amplamente utilizado.

O ESP8266 atua com uma tensão de 3,3V, tendo suporte a redes do tipo 802.11 b/g/n com um alcance máximo de noventa metros e seu sinal trabalha com 2.4 Ghz. A figura 17 ilustra o sensor de Wi-Fi.

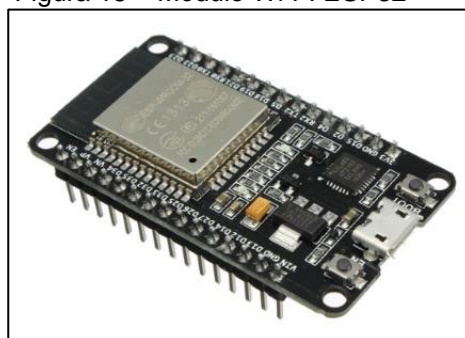
Figura 17 – Módulo Wi-Fi ESP8622



Fonte: Fingerpointengg (2018).

Há também um sensor mais sofisticado que também leva como base o módulo ESP8622, ele é o ESP32. É um módulo de alta performance com um baixo consumo de energia, com maior poder de processamento e com *bluetooth* BLE 4.2 embutido. Consta com uma interface *usb-serial* e regulador de tensão 3.3V. A programação pode ser feita em *Lua*. Com quatro *megabytes* de memória *flash*, o ESP32 permite criar variadas aplicações para projetos de IoT, acesso remoto, *webserver*s e *dataloggers*, entre outros. Entre suas especificações pode-se destacar a conexão Wi-Fi de 2.4GHz além do padrão *wireless* 802.11 b/g/ (ESPRESSIF SYSTEMS, 2013, tradução nossa). A figura 18 demonstra o modelo do módulo.

Figura 18 – Módulo Wi-Fi ESP32



Fonte: Espressif Systems (2018).

Como já descritas anteriormente as tecnologias de conexão se mostram distintas, cada qual com suas características únicas contando com vantagens e desvantagens. Vale observar ainda que para a construção do protótipo os sensores serão adquiridos e, uma vez isso abordado, mostra-se imprescindível levar em consideração os custos de cada sensor para então estar sendo adquirido.



## 5 TRABALHOS CORRELATOS

A seguir a descrição dos trabalhos utilizados na elaboração e desenvolvimento deste projeto de pesquisa

### 5.1 WEBBEE

A automação no ramo de agricultura já se trata de uma realidade. Dentro do setor da apicultura existem soluções para o monitoramento das colmeias e configura um assunto abordado com uma certa recorrência. Esse acompanhamento das abelhas por meio de sensores visa entender e descrever padrões comportamentais nas diversas situações em que o enxame estará submetido.

Existem diversas soluções no mercado e numa rápida pesquisa pode-se listar uma série de produtos com a finalidade de monitorar grandezas das colmeias, como por exemplo um sistema web denominado WebBee, apresentado por Saraiva (2003). Seu propósito é centralizar informações em diversos formatos como vídeo, imagens e textos e pesquisas realizadas na área. O WebBee conta com um módulo de sensoriamento capaz de realizar leituras de dados como temperatura, umidade e fluxo de entrada e saída das abelhas da colmeia. O projeto foi proposto a partir de uma colaboração entre o Laboratório de Abelhas do Instituto de Biociências e do Laboratório de Automação Agrícola da Escola Politécnica, ambos da Universidade de São Paulo, contando com a adesão da EMBRAPA.

### 5.2 BIG BROTHER FOR BEES (3B) – ENERGY NEUTRAL PLATAFORM FOR REMOTE MONITORING OF BEEHIVE IMAGERY AND SOUND

Murphy et al. (2015) apresentou um sistema contendo uma rede de sensores sem fio para coletar uma variedade de dados de uma colmeia afim de descrever com exatidão as condições internas e a atividade das colônias. Em seu trabalho as variáveis aferidas foram: CO<sub>2</sub>, O<sub>2</sub>, gases poluentes, temperatura, umidade relativa e aceleração. Também foram coletados dados meteorológicos para fornecer uma dimensão de análise condicional, o que garante a confiabilidade dos dados. Com uma correlação entre dados obtidos nas estações meteorológicas e os das colmeias levou

ao desenvolvimento de um algoritmo para prever a chuva a curto prazo com base nos parâmetros dentro da colmeia.

### 5.3 BEEHIVE MONITOR

Monitorar em tempo real as variantes das colmeias via Internet, tendo leituras como peso do apiário minimizando o número de visitas até ela foi o que Lewis (2014) propôs. Consiste em uma solução onde a colmeia é integrada a uma balança digital que por sua vez tem uma comunicação direta com um microprocessador de baixo custo. Esse minicomputador possui uma conexão com a Internet e dispara periodicamente um e-mail para o apicultor informando o peso da colmeia. Esta periodicidade de envio pode ser modificada conforme a necessidade. Em contraponto o sistema não salva nenhum histórico dos valores coletados para uma futura análise da produção.

### 5.4 CONTINUOUS MONITORING OF BEEHIVES' SOUND FOR ENVIRONMENTAL POLLUTION CONTROL

Pérez et al. (2016) desenvolveu um sistema para detectar alterações anormais em padrões de som proveniente das colmeias no intuito de verificar uma possível poluição química na área. Objetivando construir uma alternativa mais barata para realizar a triagem dos locais a serem analisados e, por conseguinte reduzir custos. Os dados coletados pelo sistema são gravados localmente em um cartão de memória e também enviados via General Packet Radio Services (GPRS).

### 5.5 BEE HIVE TEMPERATURE AND SOUND MONITOR

Kandepi (2015) buscou construir um hardware específico e com alta fidelidade de seus valores obtidos para monitorar a temperatura e o ruído de uma colmeia. O diferencial se encontra no modo de obter a energia para sustentar seu sistema. Os sensores são alimentados através de energia solar e enviam os dados coletados através de uma rede sem fio de curto alcance a um display LCD onde as 48

informações eram exibidas em tempo real. Porém o sistema não oferece nenhum tipo de armazenamento para esses dados coletados.

## 6 MONITORAMENTO DE COLMEIAS POR MEIO DE INTERNET DAS COISAS CONECTADO A UMA APLICAÇÃO MOBILE

Este capítulo apresenta o desenvolvimento do projeto considerando todo o referencial teórico coletado, bem como o conhecimento proveniente do mesmo. O protótipo para o monitoramento de colmeias conectado a uma aplicação *mobile* para visualização de medidas capturadas por sensores, os quais podem ser configurados em intervalos de valores máximos e mínimos para cada variável. No momento em que o valor aferido pelo sensor se encontra fora deste conjunto, a aplicação *mobile* trará um *feedback* visual alertando esta informação.

A realização da coleta dos dados, por meio dos sensores, utilizou-se de uma plataforma de prototipação *open source* popular: o Arduino, considerando o seu custo benefício. Esta ferramenta é responsável por gerenciar e processar os dados, interligando os sensores com o servidor na nuvem. Os dados posteriormente ficarão disponíveis na aplicação móvel, que foi desenvolvida utilizando-se o *framework* React Native.

### 6.1 METODOLOGIA

O método utilizado para o desenvolvimento deste projeto foi o de pesquisa quantitativa e tecnológica realizada em etapas. Primeiramente foi realizado o levantamento bibliográfico, o qual foi embasado em artigos, livros, publicações e pesquisas na Internet.

As consultas teóricas provenientes da Internet foram extraídas de bases de dados conceituadas, aferindo credibilidade a todo conteúdo descrito. Podem ser citadas as bases: Google Acadêmico; EMBRAPA relacionado a questões técnicas quanto ao cultivo de abelhas; IEEE Xplore; Elsevier e em repositórios de diversas Universidades.

Posteriormente deu-se início a escrita do referencial teórico, abordando questões primordiais para nortear e contextualizar todo o cenário apícola e as ferramentas tecnológicas necessárias que constituem assuntos importantes para analisar e desenvolver o protótipo proposto.

Esta etapa contempla temas como: apicultura e a importância das abelhas na natureza, visando compreender o impacto desse inseto na economia e em seu

habitat, além de dar ênfase nas variáveis a serem monitoradas; tecnologias para o desenvolvimento do protótipo, onde engloba Internet da Coisas e as soluções para desenvolvimento *mobile* com maior destaque para o *framework* React Native; microcontroladores e sensores para aferir as grandezas físicas ligadas a colmeia.

Com base em pesquisas e trabalhos desenvolvidos anteriormente, pode-se observar quais variáveis seriam mais adequadas para monitorar a colmeia. Em materiais disponibilizados pela EPAGRI e EMBRAPA demonstram como produzir e lidar com as abelhas visando o bem-estar animal e uma produção eficiente. As variáveis monitoradas são temperatura, umidade, peso e ruído.

A temperatura é um elemento muito importante de se observar. Independentemente da temperatura externa, a colmeia é mantida entre 34°C e 35°C, que é a ideal para desenvolvimento das crias. Caso a temperatura esteja fora desta faixa, pode provocar aumento da mortalidade na colônia e as operárias que nascerem podem apresentar defeitos físicos nas asas ou em outras partes do corpo (CAMARGO et al, 2002).

Por meio da umidade relativa podem ser detectados eventos ligados a saúde da colmeia como o aumento no consumo de alimentos ou possível abandono da colmeia (ALMEIDA, 2008). A umidade relativa no interior da colmeia é mantida por volta dos 40%. Caso o valor modifique muito por consequência da evaporação do néctar, as abelhas operárias imediatamente provocarão uma corrente de ar para o interior da colmeia, na tentativa de diminuir a umidade (CAMARGO et al., 2002).

O peso de uma colmeia reflete-se na saúde e produtividade da colônia (FITZGERALD et al., 2015). É um dado de grande utilidade para o apicultor e também pode-se obter uma estimativa da produção da colmeia de forma individual.

Padrões sonoros podem ser usados como indicadores da saúde da colmeia. O monitoramento de vibrações sonoras pode ser aplicado com estimativa do comportamento da comunidade de abelhas, identificando padrões de zumbido do enxame que indique o possível abandono em massa da colmeia (BENCSIK et al., 2011) ou possíveis acidentes através da detecção de movimentos bruscos (MURPHY et al., 2015).

As etapas consequentes estão ligadas ao desenvolvimento do protótipo em si. Tomando como partida o levantamento dos requisitos necessários, os recursos que melhor se adaptam ao cenário proposto, bem como as vantagens e desvantagens das

ferramentas partiu-se à configuração do servidor responsável por armazenar todos os dados captados dos sensores, a montagem do protótipo e testes avaliando a sua precisão, eficácias e possíveis equívocos.

Após a realização de todos esses processos descritos acima, tornou-se possível produzir a pesquisa e desenvolver o projeto/protótipo.

### 6.1.1 Ferramentas e recursos utilizados

Quanto às ferramentas utilizadas no projeto visou-se analisar o que cada tecnologia oferecia, suas vantagens, a facilidade de aprendizado levando em consideração às necessidades para o funcionamento do protótipo.

Referente ao desenvolvimento do banco de dados utilizou-se o MySQL<sup>3</sup>. Se trata de Sistema de Gerenciamento de Banco de Dados (SGBD), com a sua interface própria, *Structured Query Language* – Linguagem de consulta estruturada (SQL) – para gerenciar o conteúdo salvo. É uma tecnologia amplamente utilizada. Muitas das maiores organizações do mundo confiam e utilizam-na, incluindo Facebook, Google e Adobe, visando economizar tempo e dinheiro, abastecendo seus *sites* de alto volume de dados, sistemas críticos para os negócios (MYSQL,2019 tradução nossa).

Para a programação a nível do micro controlador Arduino foi utilizado o Arduino IDE<sup>4</sup>, versão 1.8.10, desenvolvida e fornecida pelo próprio *site* oficial do equipamento, disponibilizado gratuitamente e compatível com diferentes sistemas operacionais, como Windows, Linux e Mac OSX.

No desenvolvimento do aplicativo utilizou-se o Visual Studio Code<sup>5</sup> na versão 1.38.1 que é uma ferramenta para edição de textos completa e de grande flexibilidade, mantida pela Microsoft e disponível para diferentes tipos de sistemas operacionais. O editor de texto empregado oferece suporte para diversas linguagens de programação atuais com a vantagem de não ter custo de utilização e é uma ferramenta de código aberto.

---

<sup>3</sup> <https://www.mysql.com/>

<sup>4</sup> <https://www.arduino.cc>

<sup>5</sup> <https://code.visualstudio.com>

Referente ao desenvolvimento da aplicação *mobile* fez-se necessário algumas configurações de ambiente para a execução do *framework* React Native<sup>6</sup>. Para executar o código no computador, esta biblioteca tem dependências de algumas ferramentas, como o NodeJS<sup>7</sup> e um gerenciador de pacotes, por exemplo npm<sup>8</sup>. Além de uma abstração que facilita e agiliza todo o processo em lidar com pacotes nativos, o Expo<sup>9</sup>.

NodeJS é uma plataforma de extrema escalabilidade, trabalha em *single-thread* (única *thread* por processo) e de baixo nível. Utiliza em seu núcleo *JavaScript* como linguagem padrão de desenvolvimento. Isto só é possível pois o NodeJS utiliza a *engine JavaScript V8*, que é a mesma que o navegador Google Chrome utiliza (Pereira, 2013). Essa ferramenta vem ganhando espaço no cenário quando se trata de desenvolvimento rápido e produtivo poupando maiores problemas para os desenvolvedores.

Para gerenciar este e os demais pacotes adotou-se o NPM. O gerenciador NPM conta com uma interface via linha de comando, onde se pode adicionar e remover bibliotecas do código que está em progresso. Para programar o aplicativo foi utilizado o NPM na versão 6.4.1, sendo a mais atual no momento do desenvolvimento deste projeto. Esta ferramenta garante enorme facilidade e flexibilidade para lidar com inúmeras dependências conforme a aplicação cresce e se torna complexa.

O Expo é uma ferramenta de abstração utilizada no desenvolvimento *mobile* disponível para o React Native e é desenvolvido em JavaScript. Ela permite o fácil acesso às APIs nativas do dispositivo sem precisar instalar dependências ou lidar diretamente com código nativo (EXPO, 2019, tradução nossa). Para criar um projeto existe o recurso via linha de comando CLI onde várias funcionalidades vêm prontas, agilizando o processo de desenvolvimento.

Os testes do aplicativo foram feitos em um iPhone 6S Plus, lançado em 2015, com a versão 12.4.1 do iOS. A aplicação só é possível ser executada no *smartphone* com o aplicativo do Expo instalado. A partir dessa interação é gerado um QR Code que faz esta ponte entre os dispositivos, viabilizando os procedimentos de testes.

---

<sup>6</sup> <https://facebook.github.io/react-native/>

<sup>7</sup> <https://nodejs.org>

<sup>8</sup> <https://www.npmjs.com/get-npm>

<sup>9</sup> <https://expo.io/>

Relacionado a necessidade de armazenar os dados do protótipo, se dispôs de um servidor na nuvem. A solução se adequou perante os requisitos e necessidades analisadas previamente. Levando em consideração os custos, uma vez que soluções de servidores em nuvens habitualmente são pagas, logo buscou-se uma solução gratuita e que disponibilizasse ferramentas necessárias para hospedar os dados. O servidor encontrado que contemplou estes pontos foi o *000webhost*<sup>10</sup>. Esse servidor disponibiliza recursos como um gerenciador de arquivos, possui compatibilidade com PHP 7.1 e um banco de dados relacional MySQL com uma pré configuração e conta com um sistema de administração simples e funcional para gerenciar estas funcionalidades.

A respeito da montagem da simulação proposta para aferir os dados, foram utilizados de equipamentos listados na tabela 5.

Tabela 5 – Equipamentos utilizados para o desenvolvimento do protótipo

Descrição	Qtd	Função	Valor: US\$ Ref.: 10/2019
Placa MEGA 2560 R3 + Cabo USB para Arduino	1	Centralizar / controlar todo o fluxo de dados e demais equipamentos envolvidos	38,50
Sensor de Umidade e Temperatura DHT22	1	Capturar os valores de temperatura e umidade	9,50
Sensor de Som KY-038 Microfone	1	Capturar os valores de sonoridade	8,89
Sensor de carga	1	Capturar os valores de peso	4,71
Módulo conversor HX711	1	Converter e amplificar o sinal de 24 bits da célula de carga	6,99
Jumpers	40	Fios de conexão entre equipamentos	2,26
Protoboard 830 Pontos	1	Acoplar os sensores e disponibilizar carga necessária aos mesmos	4,30
Módulo WiFi ESP8266 ESP-01	1	Envio dos dados coletados ao servidor da nuvem	6,34
Adaptador para Módulo WiFi ESP8266 ESP-01	1	Converter a tensão de 5V do Arduino para 3.3V do módulo ESP8266	5,06
<b>Total</b>			<b>86,55</b>

Fonte: Do autor.

<sup>10</sup> <https://www.000webhost.com>

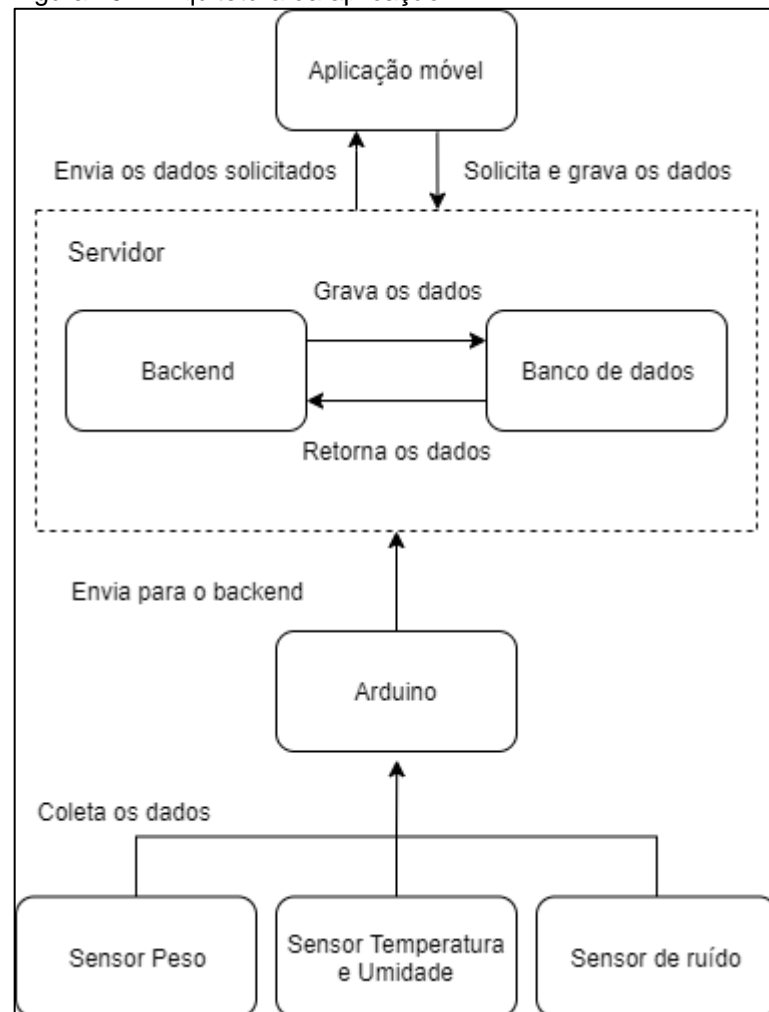


Referente aos itens demonstrados na tabela 5, tornou-se possível desenvolver o circuito responsável por gerenciar e averiguar as variáveis de ambiente relacionada a colmeia, como proposto do projeto.

### 6.1.2 Arquitetura da aplicação

O fluxo dos dados da aplicação apresentado na figura 19, é dividido em quatro elementos principais: aplicação móvel, servidor, Arduino e sensores. Cada grupo tem sua responsabilidade, inter-relacionamentos e co-dependência. O perfeito funcionamento do protótipo só se dará se cada elemento desempenhar sua função no processo.

Figura 19 – Arquitetura da aplicação



Fonte: Do Autor

A entrada do fluxo se dá pelos sensores que são responsáveis pela coleta das grandezas físicas, podem ser divididas em peso, temperatura, umidade e vibrações sonoras. Estes módulos estão conectados diretamente ao Arduino, que tem como funcionalidades centralizar os dados e processá-los para então posteriormente transmitir, via Wi-Fi através do módulo ESP8266, ao servidor na nuvem. Cada colmeia tem seu conjunto de sensores próprios que fazem a aferição em um intervalo de tempo pré estabelecido, que para fins de testes foi aplicado um intervalo de trinta segundos.

Essa conexão entre o *hardware* e o servidor é estabelecida através da Internet juntamente com um protocolo de transferência de hipertexto, do inglês *HyperText Transfer Protocol* (HTTP), servindo-se dos métodos GET e POST, conforme a etapa. O HTTP É um protocolo que permite a obtenção de recursos, tais como documentos, imagens, fotos. É a base para qualquer troca de dados na Web e um protocolo cliente-servidor, o que significa que as requisições são disparadas pelo destinatário (MDN WEB DOCS, 2019).

O GET tem como objetivo principal realizar requisições de dados a um determinado serviço diretamente via *Uniform Resource Locator* (URL), como por exemplo, o processamento de um *browser* a um *link* informado por um usuário. Já o método POST, possui uma finalidade contrária, deve ser utilizado para o envio ou submissão de dados ao servidor. Tendo como diferença também, o fato de que o POST não transmite os dados diretamente via URL, com isso não ficando visível ao cliente, no qual aumenta o nível de segurança durante a conexão (DEVMEDIA, 2012).

No projeto do monitoramento de colmeias, a transferência de dados do micro controlador ao servidor é feita pelo método POST e a transferência do servidor para o aplicativo pelo método GET.

A responsabilidade do processamento dos dados coletados é inteiramente do servidor. O *backend* foi desenvolvido em PHP, que se trata de uma linguagem de *script open source* de uso geral, altamente difundida e principalmente voltada para o desenvolvimento *web*. Com suporte para a maioria dos sistemas operacionais, ele tem como uma de suas características a possibilidade de programação estruturada bem como orientada a objetos (PHP, 2019).

Além de processar os dados, o PHP é responsável por gerenciar a conexão com o banco de dados MySQL. Essa comunicação é dada por meio de comandos

SQL, necessários para retornar as informações de acordo com o que foi solicitado pelo PHP.

Relacionado ao formato dos dados referente em cada transmissão entre as etapas do protótipo, foi padronizado o uso do *JavaScript* Object Notation (JSON). JSON é uma formatação leve de troca de dados baseado em um subconjunto de *Javascript*. É em formato de texto e completamente independente de linguagem. Se trata de um objeto, um conjunto desordenado de pares nome/valor. Um objeto começa com “{” chave de abertura e termina com “}” chave de fechamento. Cada nome é seguido por “:” dois pontos e os pares nome/valor são seguidos por vírgula (JSON, 2019)

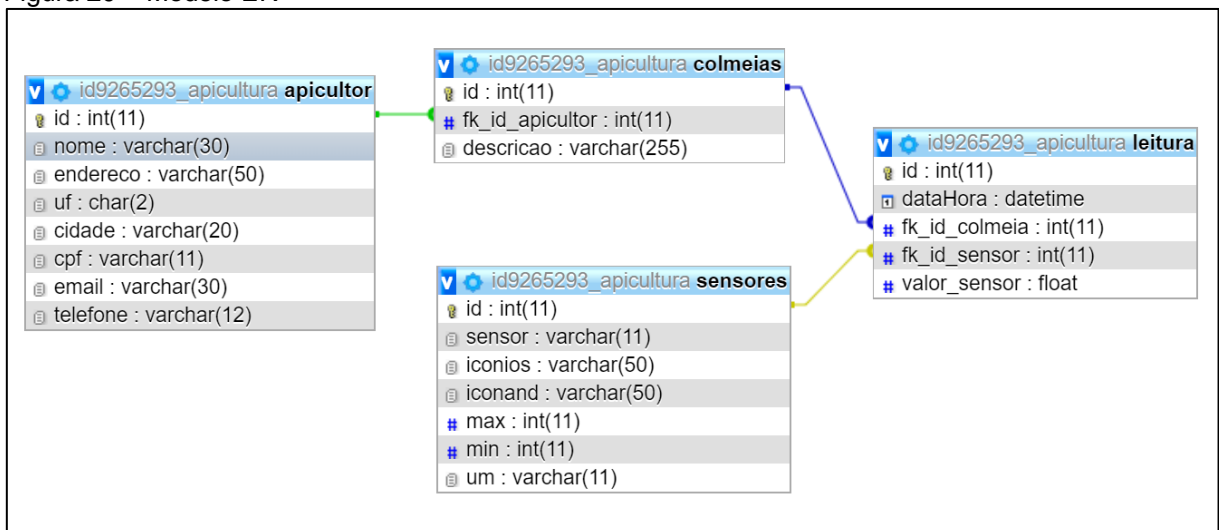
Na ponta da arquitetura encontra-se a aplicação móvel, que tem como finalidade exibir os dados aferidos pelos sensores e também estabelecer um valor mínimo e máximo para cada sensor. Caso o dado capturado esteja fora desta margem, a aplicação tem como funcionalidade demonstrar os valores para o usuário.

### 6.1.3 Diagrama de classes

O diagrama de classes referente a modelagem do banco de dados facilita o entendimento como um todo do protótipo. O seu desenvolvimento auxiliou diretamente todo o processo da criação e do fluxo dos dados, devido a capacidade que o diagrama têm em auxiliar a visualização do conjunto.

A figura 20 ilustra o modelo lógico do banco de dados presente no projeto.

Figura 20 – Modelo ER



Fonte: Do autor.

Conforme demonstrado na figura 20, o banco de dados compreende quatro tabelas, que são:

- a) apicultor: têm a responsabilidade de armazenar os dados referente ao apicultor, o proprietário da colmeias. Algumas informações como o seu nome, endereço, cidade, CPF, e-mail e telefone;
- b) colmeias: uma tabela intermediária que tem um relacionamento entre apicultor e leitura, podendo diferir qual leitura de uma colmeia pertence a qual proprietário;
- c) leitura: responsável por armazenar todos os dados coletados, de forma individual de cada sensor e também um histórico tendo data e horário de cada aferição;
- d) sensores: é armazenado o cadastro de cada sensor utilizado no projeto.

Com a modelagem concluída do que se trata de banco de dados, foi realizado um estudo referente as ferramentas disponíveis e que se adequariam as necessidades do projeto, para ser utilizado no desenvolvimento e montagem do protótipo.

#### **6.1.4 Sensoriamento e processamento do Arduino**

Para o desenvolvimento do protótipo foram utilizados 4 sensores, DHT22 (temperatura e umidade), KY-038 (vibrações sonoras), célula de carga (peso) e o ESP8622 (Wi-Fi) e uma *protoboard*. Para os funcionamentos da célula de carga e do Wi-Fi foram necessários adaptadores permitindo que o Arduino mantivesse a autonomia de gerenciá-los. Para a célula de carga foi acoplado um módulo conversor, o HX711. E referente ao ESP8622 foi empregado um adaptador para lidar com o sinal de 5V que é a tensão compatível do Arduino. Esse modelo de adaptador para Wi-Fi tem como vantagem a sua quantidade de pinos que facilita a conexão na *protoboard*, reduzindo o uso de fios no protótipo.

Iniciou-se a montagem do protótipo pela *protoboard*, tendo este componente a responsabilidade de ser a base para todas as conexões entre sensores e o Arduino. O modelo utilizado tem como características principais 830 furos, podendo operar com uma tensão máxima de 500V corrente alternada (AC) por minuto e uma faixa de temperatura variando de -20°C a 80°C.

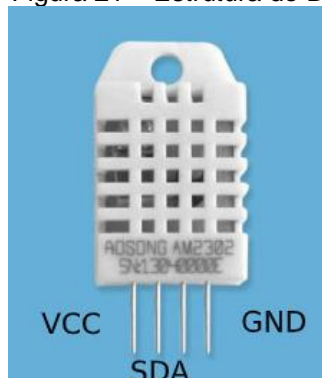
As etapas seguintes são referentes a montagem física dos sensores, conexão na *protoboard* e na plataforma de prototipação. Posteriormente com os sensores conectados procedeu-se com a programação na IDE disponibilizada pelo próprio fabricante, que no caso é o Arduino.

O módulo DHT22 tem como característica aferir as grandezas temperatura e umidade. Segundo os dados fornecidos em seu *datasheet* disponibilizados pela fabricante Aosong (2019, tradução nossa), o sensor possui a capacidade de coletar dados que estão em uma variação de temperatura entre  $-40^{\circ}\text{C}$  a  $80^{\circ}\text{C}$  e quanto a umidade relativa os valores compreendidos entre 0% e 99,9%.

Outra característica desse sensor é sua capacidade de atuar entre valores de 3,3V até 5V, embora o recomendado seja 5V. Ele dispõe de um controlador de 8 bits que é responsável por gerenciar os dois dispositivos sensíveis – umidade e temperatura – o que lhe dá alta confiabilidade e excelente estabilidade a longo prazo. É um sensor pequeno, com baixo consumo de energia e atua com uma distância de sinal de até 20 metros (AOSONG, 2019, tradução nossa).

O DHT22 é composto por quatro terminais, vide figura 21, ao qual destes apenas três são utilizados para seu funcionamento no protótipo. A primeira conexão é o VCC que é conexão responsável por alimentar o sensor com a voltagem de corrente contínua; o segundo é o SDA e atua na transmissão dos dados coletados e conectado a uma porta analógica do micro controlador e o último terminal é o GND, que se constitui o terra.

Figura 21 – Estrutura do DHT22

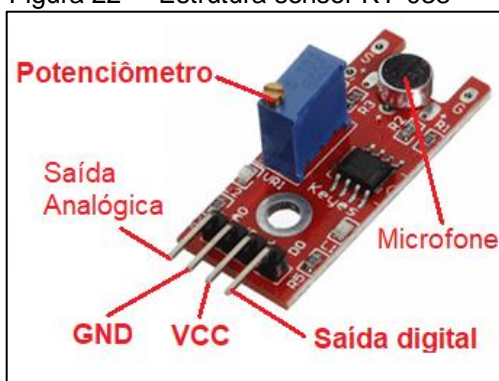


Fonte: Adaptado de Aosong (2019).

Para a coleta das vibrações sonoras utilizou-se o sensor KY-038. Ele é dotado de um microfone em uma de suas extremidades com um pequeno amplificador

embutido para envio dos dados captados. A detecção de som gera uma saída que é percebida pelo Arduino através da variação da tensão (uma menor tensão indica que há som). Este sensor é muito utilizado para protótipos de automação residencial e iluminação de ambientes, visto que este modelo, além de uma saída analógica, também possui uma saída digital (ALVENS, MORAIS, 2018). A figura 22 demonstra o sensor em questão

Figura 22 – Estrutura sensor KY-038



Fonte: Adaptado de Datasheetcafe (2018).

O KY-038 tem em sua estrutura a presença de quatro terminais, conforme figura 21, que são: saída analógica, terra, alimentação e saída digital. Ele também possui um potenciômetro para controlar manualmente a sensibilidade a qual o microfone irá captar as vibrações sonoras. Com a detecção de algum ruído há uma confirmação visual feita por meio de dois LEDs de tensão e da saída digital.

Após estabelecidas as conexões necessárias do sensor com o Arduino, os valores providos pela porta analógica serão em formato numérico, representando a variação do som que está ocorrendo no ambiente. Já em relação à porta digital, os valores obtidos variam de acordo com as detecções sonoras realizadas, onde para cada identificação efetuada, o equipamento alterna o estado de sua saída digital entre 0 e 1 (MASTERWALKER, 2018).

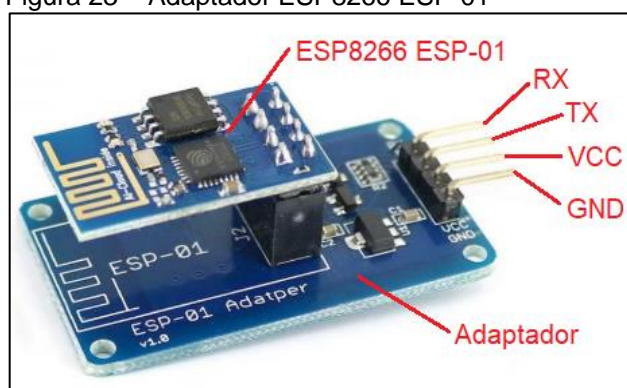
Com essa operação obteve-se uma maximização nas conexões dos terminais, pois passou-se a manusear com quatro terminais ao invés dos oito originais disponíveis no sensor.

Segundo as informações contidas no *datasheet* disponibilizado pela Espressif Systems (2013, tradução nossa), este componente tem objetivos prover conexão com a Internet via Wi-Fi de forma eficaz e de baixo custo e ser capaz de

operar nas redes 802.11 b/g/n, que pode atuar como um ponto de acesso ou no formato de uma estação, enviando e recebendo informação a uma distância aproximada de 90 metros de acordo com cada ambiente (ESPRESSIF SYSTEMS, 2013, tradução nossa). A tensão de operação é de 3.3V, suportando as comunicações do tipo TCP (do inglês, *transmission control protocol*) e UDP (do inglês, *user datagram protocol*), com até cinco conexões simultâneas, tendo a sua comunicação com o Arduino no formato serial, efetivada pelos pinos TX e RX por meio de comandos AT (ESPRESSIF SYSTEMS, 2013, tradução nossa).

Com esse adaptador acoplado ao sensor, diminui a quantidade de pinos para trabalhar e também tem o recurso de converter a tensão original do Arduino de 5V, para a tensão de operação do módulo Wi-Fi de 3.3V. É composto por quatro terminais que são: TX e RX, utilizados para a transmissão de dados, VCC e GND que estão relacionados a alimentação do sensor. A figura 23 ilustra este conjunto e seus terminais.

Figura 23 – Adaptador ESP8266 ESP-01



Fonte: Adaptado de StackExchange (2018).

Entre todas as características já salientadas previamente referente a utilização do módulo Wi-Fi, destaca-se a possibilidade de realizar o monitoramento dos dados coletados em tempo real, eliminando a necessidade de estar presente no local das colmeias em questão.

Outra vantagem no uso desta arquitetura diz respeito à confiabilidade e segurança dos dados. Logo após a leitura dos sensores, os dados são enviados para um servidor remoto, que no projeto se localiza na nuvem, via Internet, trazendo maior robustez para o protótipo, tendo em vista que um servidor local poderia trazer inúmeros problemas se for agregado direto na área de produção apícola.

As aferições são feitas dentro de um espaço-tempo regular e constante, acumulando dados e históricos. O servidor em nuvem é escalável e garante o armazenamento de dados de forma contínua e crescente. No caso de um servidor local, as soluções de armazenamento de dados e a forma de comunicação, poderiam acarretar em maiores custos em relação ao protótipo apresentado.

O sensor de carga (figura 24), responsável por medir o peso da colmeia, é um Strain Gauge. Trata-se de um medidor de tensão capaz de averiguar as alterações nas resistências elétricas em resposta a tensão (ou pressão ou força) e de forma proporcional aplicada ao dispositivo. O Strain Gauge mais comum é constituído por fio ou folha muito fina, configurada em um padrão de grade, de modo que haja uma mudança linear na resistência elétrica quando a tensão é aplicada em uma direção específica (AL-MUTLAQ, 2018, tradução nossa).

A sensibilidade do sensor geralmente pode ser expressa pelo chamado Gauge Factor, GF. É calculado a partir da razão entre uma mudança na resistência elétrica em relação a uma mudança no comprimento provocada por uma força externa. A peculiaridade neste sensor é que uma força ou pressão externa muito pequena provoca uma variação na escala de décimos de ohms. Devido a essas variações de ordem muito pequena é utilizado um amplificador de sinais (VIDAL, 2018).

Figura 24 – Sensor de peso



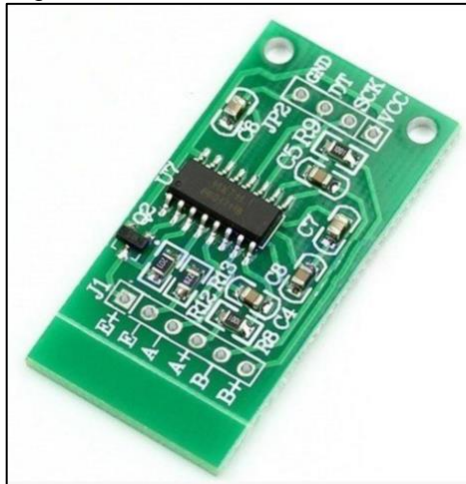
Fonte: Vidal (2018)

Utilizou-se o amplificador HX711 para tratar esses pequenos valores provenientes da célula de carga. Este módulo, segundo o *datasheet* disponibilizado pela fabricante Avia Semiconductors, é um conversor analógico-digital (ADC) de precisão de 24 bits projetado para balanças e aplicações de controle industrial para



interface diretamente com um sensor de ponte. A figura 25 demonstra o amplificador com suas respectivas conexões.

Figura 25 – Módulo conversor HX711



Fonte: Vidal (2018)

O módulo foi desenhado para interfacear diretamente com sensores em ponte para aplicações de medição de carga. O multiplexador de entrada seleciona entre dois canais diferenciais A e B. Cada canal diferencial pode ser ligado em até duas células de carga – A+ e A- e B+ e B-, cada qual ligado na saída de uma célula de carga em meia ponte (VIDAL, 2018).

Com os dispositivos devidamente energizados e conectados, a responsabilidade de orquestrar todos os sensores é do Arduino. No desenvolvimento do protótipo utilizou-se o modelo Mega 2560 para gerenciar a aplicação.

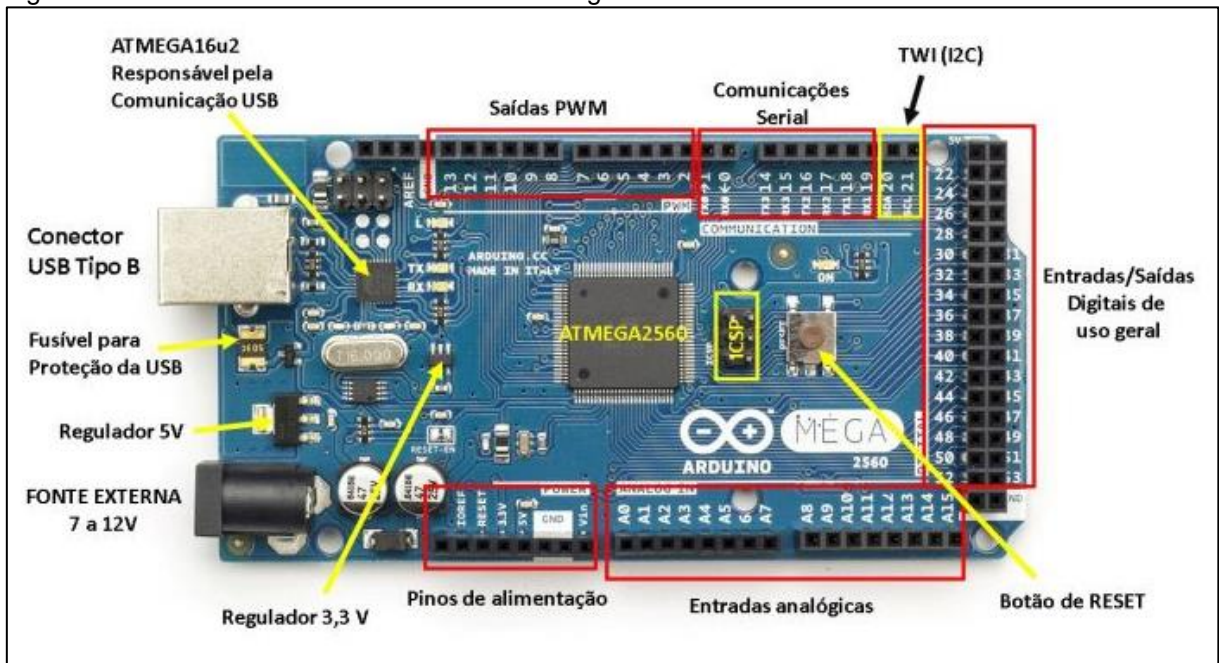
De acordo com o *site* oficial da plataforma (2019, tradução nossa), o Arduino Mega 2560 é um microcontrolador baseado na placa ATmega2560. Contém 54 pinos digitais, dos quais 15 podem ser utilizados como saídas PWM, 4 UARTs (portas seriais de hardware), 16 entradas analógicas, uma conexão USB, um oscilador de cristal de 16 MHz, um conector ICSP, um conector de alimentação e um botão *reset*.

Cada um dos 54 pinos digitais que o Mega contém, podem ser utilizados tanto como entrada, como para saída, podendo-se utilizar as funções *pinMode*, *digitalWrite* e *digitalRead*. Todas estas entradas operam com uma tensão de 5V,

podendo fornecer e receber 20mA como condição de operação recomendada (ARDUINO, 2019, tradução nossa).

A figura 26 ilustra o conjunto completo referente a estrutura do Arduino Mega, como citado anteriormente.

Figura 26 – Resumo dos recursos do Arduino Mega2560



Fonte: Souza (2014).

A principais características a respeito do Arduino dizem respeito ao baixo custo para sua aquisição e ao *software* de código aberto. Entretanto destaca-se também como seus diferenciais decisivos para sua escolha para o projeto, a existência de IDE própria para o desenvolvimento dos códigos, de fácil aprendizagem e compatibilidade com os diversos sistemas operacionais.

Referente ao modelo, Arduino Mega 2560, optou-se devido as suas especificações técnicas, onde foram considerados o poder de processamento, a quantidade de saídas e entradas, por se constituir modelo robusto, já que se objetivou um protótipo escalável ao decorrer do tempo.

### 6.1.5 Ambiente de processamento

De acordo com o tópico referente às ferramentas e recursos, no que se trata de tecnologias do servidor e armazenamento de dados, optou-se pela plataforma

gratuita *000webhost*, o qual demonstrou tratar-se de um serviço no qual atendeu todos os requisitos exigidos para o funcionamento do protótipo.

O desenvolvimento do *backend* deu-se a partir da modelagem e da estrutura do banco de dados MySQL. As tabelas e seus respectivos inter-relacionamentos foram criadas por meio de uma interface disponibilizada pelo servidor aliada à ferramenta PhpMyAdmin. A interface permite visualizar as tabelas, suas colunas e seus relacionamentos e facilitar a compreensão e a visualização do banco de dados como um todo. A figura 27 ilustra o painel contendo as ferramentas para o gerenciamento das tabelas do banco de dados.

Figura 27 – Tabelas do banco de dados



Tabela	Acções	Registos	Tipo	Agrupamento (Collation)	Tamanho	Suspensão
apicultor	Procurar, Estrutura, Pesquisar, Inserir, Limpar, Eliminar	2	InnoDB	utf8_unicode_ci	16 KB	-
colmeias	Procurar, Estrutura, Pesquisar, Inserir, Limpar, Eliminar	2	InnoDB	utf8_unicode_ci	32 KB	-
leitura	Procurar, Estrutura, Pesquisar, Inserir, Limpar, Eliminar	1,134	InnoDB	utf8_unicode_ci	96 KB	-
sensores	Procurar, Estrutura, Pesquisar, Inserir, Limpar, Eliminar	4	InnoDB	utf8_unicode_ci	16 KB	-
4 tabelas	Soma	1,142	InnoDB	utf8_unicode_ci	160 KB	0 Bytes

Fonte: Do autor

Para gerenciar a estrutura fez-se uso de uma linguagem padronizada universalmente que gerencia esse tipo de dados: o SQL. A figura 28 exemplifica um trecho de código utilizado para criação da tabela “apicultor” presente no protótipo.

Figura 28 – Comando usado para criar a tabela apicultor

```
CREATE TABLE `apicultor` (
  `id` int(11) NOT NULL COMMENT 'ID Apicultor',
  `nome` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `endereco` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `uf` char(2) COLLATE utf8_unicode_ci NOT NULL,
  `cidade` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `cpf` varchar(11) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(30) COLLATE utf8_unicode_ci NOT NULL,
  `telefone` varchar(12) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Fonte: Do autor

A partir do banco de dados devidamente estruturado, a interface do PHPMYADMIN disponibilizada pelo servidor inúmeras ferramentas para a manipulação da base de dados. Pode-se inserir dados e efetuar consultas individualmente em cada tabela, tanto por meio de uma interface gráfica quanto com

comandos em SQL. Na figura 29, utilizando um comando SQL, pode-se visualizar registros da tabela. Este simples comando irá retornar todos os dados contidos na tabela *apicultor*.

Figura 29 – Selecionando dados da tabela *apicultor*

```
SELECT * FROM 'apicultor'
```

Fonte: Do autor

Para efetivar a comunicação entre o Arduino e a base de dados foi utilizado *script* desenvolvido em PHP, uma vez que o servidor tem compatibilidade para esta linguagem de programação.

A estrutura padrão para o desenvolvimento de *scripts* baseados na linguagem PHP é sinalizado com a abertura e fechamento da *tag* `<?php` e `php>`, e o bloco de código inserido entre está *tag* contém a lógica de programação necessária para processar os dados. A fim de estabelecer uma conexão do banco de dados com o servidor utilizou-se de funções que tratam exclusivamente com o relacionamento entre PHP e banco de dados.

A conexão com o banco de dados foi obtida através do método *mysqli\_connect*, o qual permite enviar, por meio de parâmetros, algumas configurações como localização do banco, usuário, senha e o nome da base de dados. A localização do banco de dados é determinada pelo *localhost*, devido ao PHP estar situado no mesmo ambiente que o banco. Já as configurações de usuário, senha e nome da base são definidas no momento em que o banco é criado no servidor. Na figura 30 é demonstrado a conexão e seus parâmetros necessários.

Figura 30 – Conexão banco de dados

```
$connect = mysqli_connect(
    "localhost",
    $user,
    $senha,
    $banco
);
```

Fonte: Do autor

Com a conexão estabelecida utiliza-se a função *mysqli\_query* com o intuito de manipular os dados, bem como as tabelas propriamente ditas do banco, possibilitando o acesso a recursos como consultas, alterações e exclusões de

registros. Por meio desta função são passados os dados de acesso ao banco, definidos anteriormente pela função *mysqli\_connect* e o comando SQL que se deseja executar. A figura 31 ilustra a montagem de uma *query* referente a tabela “apicultor”, onde ela vai buscar um único registro baseada no e-mail cadastrado.

Figura 31 – Comando para buscar registro

```
$query = 'select apicultor.id,
            apicultor.cidade,
            apicultor.cpf,
            apicultor.email,
            apicultor.endereco,
            apicultor.nome,
            apicultor.telefone,
            apicultor.uf
        from apicultor where apicultor.email = "'.$jsonObj->email.'";
$result = mysqli_query($connect,$query) or die('{"verro":"true","vretorno":"ERRO NO COMANDO SQL DO APICULTOR"}');
$apicultor = "";
```

Fonte: Do autor

Para obtenção dos dados da pesquisa, pode ser usado a função *mysqli\_fetch\_assoc*. Esta funcionalidade tem como resultado um *array* que contém os dados que foram solicitados via *query*.

Figura 32 – Consulta tabela

```
while($retorno=mysqli_fetch_assoc($result)){
    if (!empty($apicultor))
        $apicultor = $apicultor.',';

    $apicultor = $apicultor.'{"id":"' . $retorno['id'] . '","cidade":"' . $retorno['cidade'] .
        '","cpf":"' . $retorno['cpf'] . '","email":"' . $retorno['email'] . '","endereco":"' . $retorno['endereco'] .
        '","nome":"' . $retorno['nome'] . '","telefone":"' . $retorno['telefone'] .
        '","uf":"' . $retorno['uf'] . '"}';
}
```

Fonte: Do autor

O método *mysqli\_close* é responsável por finalizar a comunicação entre o PHP e o banco de dados. É uma função necessária para evitar possíveis erros nas consultas de dados.

A figura 32 exemplifica o processo de consulta em uma tabela, representada pela tabela “apicultor”, retornando todos os dados presente nas colunas.

O *script* PHP é responsável pela centralização da parte lógica do protótipo. Nele constam as regras do sistema e as questões de maior relevância dos dados aferidos.

A comunicação através das diretivas do HTTP, no protótipo foram desenvolvidos dois métodos disponíveis neste protocolo: o GET e o POST, que foram especificados anteriormente. Partindo do recebimento dos dados obtidos pelos sensores e na sequência identificar o tipo de operação proveniente da aplicação

móvel, ambas em formato JSON, é possível determinar qual método irá ser processado a partir da demanda do usuário.

Após a conclusão do desenvolvimento dos códigos utilizando o PHP, foi realizado o seu *upload* para o servidor. Utilizou-se uma ferramenta disponibilizada pelo *000webhost* que facilitou e agilizou este processo referente ao envio do código para o servidor.

### 6.1.6 Desenvolvimento da aplicação móvel

A última etapa do desenvolvimento do projeto foi a programação referente a construção do aplicativo. Utilizou-se um *framework* híbrido, React Native, afim de cobrir a maior parte dos sistemas operacionais embarcados nos *smartphones* atualmente.

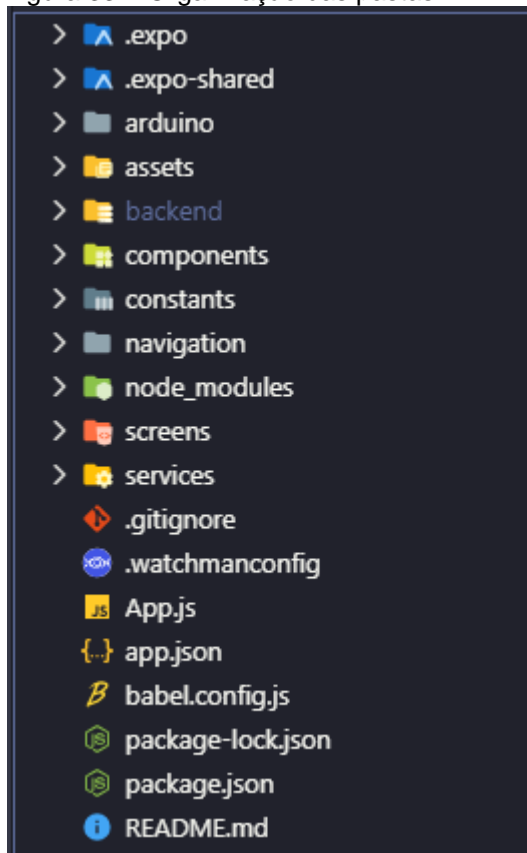
Dentro das características do *framework*, observou-se o seu desempenho em multiplataforma, comparado com o desenvolvimento nativo. O React Native é uma tecnologia desenvolvida pelo Facebook e de código aberto, conta com uma comunidade entusiasta, facilitando o processo de aprendizagem. O fato de ser multiplataforma tem a vantagem relacionada ao tempo, tornando o desenvolvimento ágil em um curto prazo.

O Expo foi utilizado com o intuito de facilitar o desenvolvimento, com suas ferramentas e abstrações prontas para o uso, uma vez que ele minimiza a complexidade de lidar com questões nativas, além de ser uma forma mais fácil para interagir com o React Native.

Para desenvolver usufruindo dessas ferramentas foi aplicada a linguagem *JavaScript*. Trata-se de uma linguagem de programação que permite a criação de conteúdos que se atualizam dinamicamente, controlam multimídias, imagens animadas, entre outros recursos (MDN, 2019). O *JavaScript* é uma linguagem amplamente utilizada, versátil e de fácil aprendizado e vem sendo amplamente utilizada os desenvolvedores.

No desenvolvimento do aplicativo foram usados recursos da CLI – Command Line Interface disponibilizados pelo próprio Expo. Com essa linha de comando permite criar um projeto previamente configurado, fator que acaba economizando tempo e possibilita uma maior produtividade (figura 33).

Figura 33 – Organização das pastas



Fonte: Do autor

O código já vem com a estrutura de pastas e não exigiu nenhuma alteração nos diretórios para o desenvolvimento do aplicativo. Para a estilização da interface, as telas que o usuário tem contato direto, e utilizou-se *styled-components*. Essa biblioteca é compatível com o React Native e permite utilizar estilos ao nível de componente na sua aplicação. Os estilos são escritos em uma mistura de *JavaScript* com CSS (RABELO, 2017).

A parte lógica do *frontend* vincula os dados vindos do servidor e é de responsabilidade dos arquivos em *JavaScript*. A maior parte dos códigos do aplicativo foi desenvolvida utilizando esta linguagem de programação. Toda a estrutura de arquivos e relacionamentos dentro da aplicação se dá através dos conceitos do React Native.

A inicialização do aplicativo parte do arquivo *app.js*, que se encontra na raiz da pasta do projeto, e é responsável por gerenciar qual imagem será exibida quando o aplicativo estiver carregando e a primeira tela que será mostrada, além de questões de estilização, como a fonte dos textos.



A parte de navegação entre as telas está dentro da pasta *navigation*. Nela estão contidos dois arquivos onde são descritas todas as telas e o fluxo de navegação. As telas estão localizadas dentro da pasta *screens*. Cada tela tem o seu respectivo arquivo *JavaScript*.

O desenvolvimento do aplicativo se orientou por um padrão de códigos que respeitou a arquitetura e dividiu as responsabilidades em arquivos distintos visando maximizar questões pertinentes à manutenção e a compreensão do mesmo.

Os dados vindos da API são manipulados internamente em cada *screen*. Para estabelecer uma conexão entre a aplicação e o *backend* e a comunicação com as requisições baseadas no protocolo HTTP, por meio de *request* e *response*, foi utilizada a biblioteca *axios*.

Na aplicação foram empregados dois métodos disponíveis do HTTP, no caso são eles: GET e POST. Devido a suas características e para manter um padrão de código. Cada requisição tem seu retorno, e quando se utiliza o GET, o retorno que foi desenvolvido é formatado no padrão JSON.

Com os dados vindo da API (figura 34), provenientes de uma chamada a Uniform Resource Locator, em português é conhecido por Localizador Padrão de Recursos (URL), a responsabilidade de exibir estas informações no aplicativo é do *frontend* em si. Na manipulação desses objetos vindos através JSON do *backend* utilizou-se laços de repetições, verificações, condicionais e uma série de funções lógicas tanto para validar quanto para exibir.

Figura 34 – Dados vindo da API

```
return this.state.leituras.length ? (
  <View style={styles.container}>
    {this.state.leituras.length > 0 ? this.state.leituras.map(leitura => (
      <TouchableOpacity key={leitura.id}
        onPress={() => {this.props.navigation.navigate
          ('Log', { title: 'Histórico',
            id: `${leitura.any}`,
            um: `${leitura.um}`,
            title: `${leitura.sensor}` }) } })
        style={styles.button}>
          <View style={styles.info}>
            <View style={{flexDirection: 'row', alignItems: 'center'}}>
              <TabBarIcon
                style={styles.buttonIcon}
                name={Platform.OS === 'ios' ? `${leitura.iconios}` : `${leitura.iconand}`} />
              <Text style={{color: `${leitura.cor}`}}> {leitura.sensor}</Text>
            </View>
            <Text style={{color: `${leitura.cor}`}}> {leitura.valor_sensor} {leitura.um}</Text>
          </View>
          <View style={styles.date}>
            <Text style={{color: '#999'}}>{format(parseISO(leitura.dataHora), "dd 'de' MMMM', às 'HH:mm'h'", { locale: pt })}</Text>
          </View>
        </TouchableOpacity>
      )}
    )}
  </View>
```

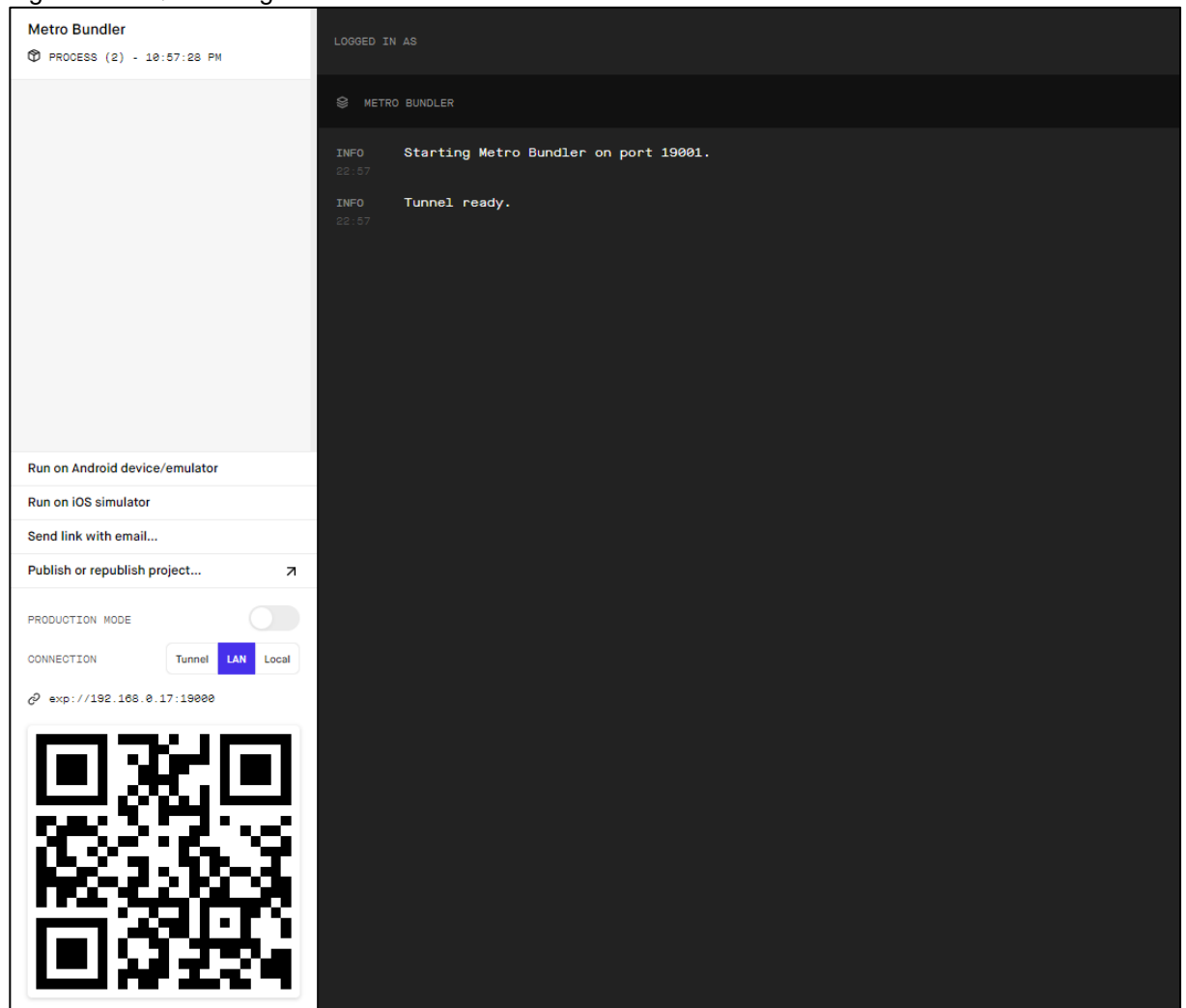
Fonte: Do autor



Observa-se que está comunicação entre o servidor na nuvem e a aplicação depende de ambos estarem conectados à Internet consistindo em uma solução *online* que disponibiliza os valores praticamente em tempo real, dispondo de segurança já que os dados estarão hospedados em um servidor remoto e disponíveis para gravar e alterar a qualquer instante.

Por meio do comando *npm start* dentro da pasta do projeto, o Expo se encarrega de disponibilizar um servidor e um QR Code. Ao escanear o QR Code – é necessário que o aplicativo Expo esteja instalado, independente do sistema operacional do celular – a aplicação será carregada no dispositivo físico.

Figura 35 – QR Code gerado



Fonte: Do autor

Conforme figura 35, o QR Code é gerado e então a aplicação poderá ser executada em um *smartphone*. Assim o *feedback* referente as questões de

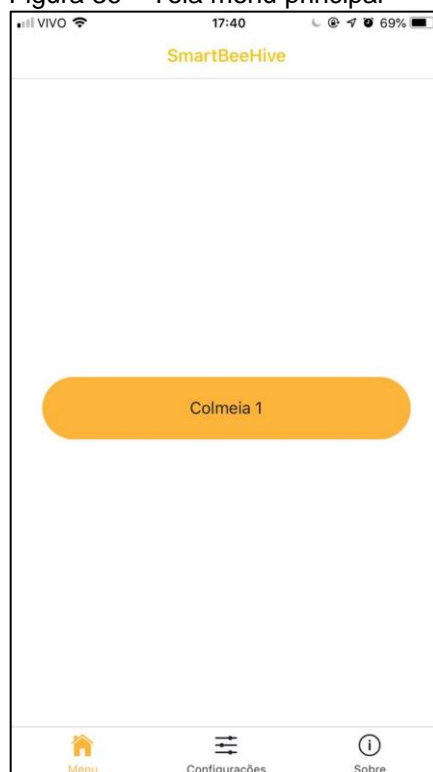
usabilidade e design são mais concretas se comparadas a execução da aplicação em um celular emulado no próprio computador.

O código do aplicativo se torna mais complexo a cada nova funcionalidade desenvolvida. A fim de lidar com esta escalabilidade, aplicou-se o conceito de versionamento de código. Utilizou-se o Git para o versionamento e o GitHub<sup>11</sup> como plataforma para hospedar o código fonte. Além de uma boa prática de programação, o código se mantém sempre na última versão, caso necessário poderia regredir a uma versão anterior como por exemplo, se ocorrer algum erro de implementação.

### 6.1.7 Aplicação em funcionamento

Após realizadas as etapas de desenvolvimento do protótipo partiu-se para a fase de demonstração do funcionamento do aplicativo. A aplicação é composta por uma série de telas, cada uma com sua finalidade específica, organizadas em formato de menu, conforme a figura 36 mostrada abaixo.

Figura 36 – Tela menu principal



Fonte: Do autor.

---

<sup>11</sup> <https://github.com/tiagobehenck/smartbeehive>

A funcionalidade “menu” é a primeira opção intuitiva do menu, localiza-se no canto inferior esquerdo e permite a navegação de retorno à tela principal. No centro do aplicativo tem-se a relação de colmeias monitoradas e cadastradas no banco de dados. As colmeias podem ser dispostas no visor em ordem crescente de acordo com a identificação única de cada unidade. Ao selecionar o ícone de uma colmeia, o aplicativo é direcionado para a tela onde são mostrados os dados coletados pelos sensores (figura 37).

Figura 37 – Tela dos dados dos sensores



Fonte: Do autor

Os dados são apresentados em fonte na cor preta quando os resultados das amostras estão em conformidade aos valores esperados, de acordo com o pré determinado pelo usuário do aplicativo para cada grandeza (umidade, temperatura, peso e ruído) em determinada colmeia. Quando alguma dessas grandezas apresentar valores nas medições fora do intervalo ideal estabelecido, a fonte é exibida na cor vermelha. A tela também identifica a data e hora em que cada dado foi coletado e consta no canto inferior direito no campo de cada item monitorado. Observa-se que a temperatura é dada em graus Celsius (°C), a umidade em percentual relativo do ar (%), o peso em quilogramas (kg) e o ruído em decibéis (dB).

Ao selecionar um dos itens coletados, a aplicação possibilita ao usuário um histórico dos dados, com opção de intervalo de tempo, das medições realizadas no período selecionado (figura 38). Esse detalhamento é uma relação dos dados coletados em ordem decrescente de data e horário.

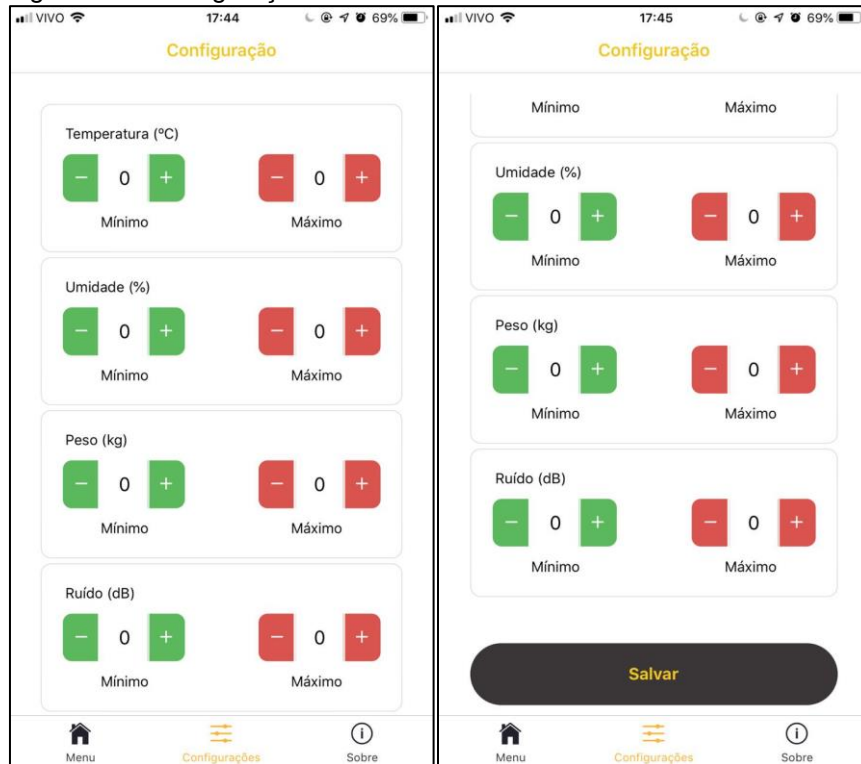
Figura 38 – Tela de histórico dos sensores

Temperatura	
27 de setembro, 22:18:59	21.5 °C
27 de setembro, 22:18:42	21.5 °C
27 de setembro, 22:18:25	21.6 °C
27 de setembro, 22:18:08	21.5 °C
27 de setembro, 22:17:50	21.5 °C
27 de setembro, 22:17:33	21.4 °C
27 de setembro, 22:17:16	21.4 °C
27 de setembro, 22:16:59	21.4 °C
27 de setembro, 22:16:42	21.4 °C
27 de setembro, 22:16:25	21.4 °C

Fonte: Do autor

Retornado ao menu principal o usuário dispõe da opção “configurações” no centro inferior da tela. Salienta-se que um ícone se difere dos demais por apresentar-se em fonte na cor amarela quando selecionado. Em “configurações”, o usuário da aplicação tem a possibilidade de configurar os intervalos de valores mínimos e máximos para cada variável. Essa opção está representada na figura 39 abaixo.

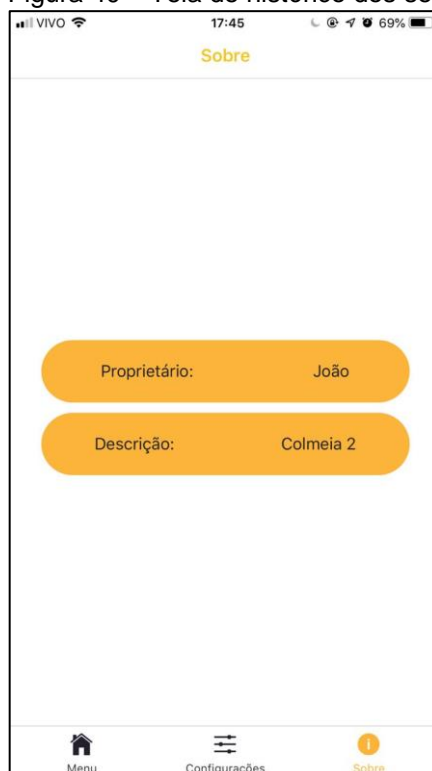
Figura 39 – Configuração de mínimo e máximo dos sensores



Fonte: Do autor

A variável temperatura permite que a configuração de valores mínimos de zero grau Celsius e máximo de cem graus Celsius. Os botões de ajuste, divididos entre o sinal de menos “-”, à esquerda do campo para dígitos, e o sinal de mais “+”, à direita do campo para os dígitos, são na cor verde para os valores mínimos e na cor vermelha para os valores máximos. O mesmo padrão de cores e valores ocorre nas variáveis umidade, peso e ruído.

Figura 40 – Tela de histórico dos sensores



Fonte: Do autor

Finalmente, no menu principal, há a opção “sobre” no canto inferior direito da tela. Essa função identifica o proprietário da colmeia e a descrição que identifica a unidade produtiva de que tratam as colmeias que estão sob monitoramento (figura 40 – foto da opção “sobre”). As informações contidas na aba “sobre” são previamente inseridas no aplicativo através de cadastro diretamente no banco de dados pelo desenvolvedor.

#### 6.1.8 Testes do projeto

Realizada a conclusão de todas as etapas referente ao desenvolvimento do projeto, foi possível realizar testes para validar a precisão e acurácia dos sensores, o poder do processamento, a capacidade de armazenamento e a visualização dos dados.

Para o efetivo funcionamento do aplicativo, inicialmente se faz necessário configurar o banco de dados, de forma manual, inserindo as informações referentes a identificação da colmeia a ser monitorada, os parâmetros dos sensores e o cadastro do apicultor. É imprescindível identificar o tipo de sensor através de uma chave única

para estabelecer um relacionamento com as demais tabelas, possibilitando a sua identificação na tabela “leituras”, por exemplo. A mesma lógica é aplicada na tabela de colmeias, tornando possível identificar a qual proprietário ela pertence.

Com as devidas tabelas preenchidas é possível inicializar o protótipo a fim de testá-lo, observando como cada componente irá se comportar em diferentes situações estipuladas.

Os testes foram realizados de forma individual em cada sensor, buscando visualizar possíveis alterações nos valores aferidos. Essa atividade de análise do protótipo partiu da conexão do Arduino em uma rede de Wi-Fi com a finalidade de se comunicar e transferir os dados para o servidor.


























































O sensor de temperatura e umidade foi estressado utilizando um secador de cabelo, regulado para produzir ar quente numa velocidade média, no qual se pode observar que os valores imediatamente responderam às alterações provocados pelo experimento. A temperatura aferida no aplicativo aumentou e houve variação do valor da umidade relativa do ar.

A testagem do sensor de ruído foi feita em dois cenários opostos: um silencioso e outro com barulho. Na exposição do sensor às duas situações, observou que – indiferente da configuração realizada manualmente no potenciômetro responsável pela sensibilidade do microfone – os valores não demonstraram variações. Levantou-se a hipótese de que o sensor empregado estava com alguma avaria, algum defeito nesse sensor em específico e não no modelo em si.

Para analisar a carga, o sensor de peso em si, foi aplicada uma pressão com o intuito de aferir a variação decorrente da deformação da sua resistência. Embora tenha sido calibrado devidamente e testado diferentes formas de desenvolvimento de código, o sensor não demonstrou precisão quando foi sobreposto um objeto de peso conhecido.

A figura 41 demonstra os dados coletados presentes no banco de dados referente aos testes realizados.

Figura 41 – Tabela leituras com os dados

			dataHora	fk_id_sensor	valor_sensor				
			Data e Hora	ID do sensor	Valor do sensor lido				
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:14	1	22.1
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:14	2	82
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:14	3	18
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:14	4	503688
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:31	1	22.5
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:31	2	81.4
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:31	3	18
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:31	4	508516
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:48	1	22.5
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:48	2	81.3
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:48	3	18
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:05:48	4	508734
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:05	1	22.5
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:05	2	81.1
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:05	3	18
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:05	4	509188
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:23	1	22.5
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:23	2	80.9
<input type="checkbox"/>		Edita		Copiar		Apagar	2019-11-05 20:06:23	3	18

Fonte: Do autor

O Arduino operou normalmente gerenciando todos os sensores acoplados, realizando as coletas de dados e enviando-os para o servidor em questão, demonstrando nenhum equívoco.

O aplicativo foi testado em um *smartphone* dispondo do sistema operacional iOS na versão 12.4.1, no modelo iPhone 6S Plus da Apple. Com isso analisou-se todas as funcionalidades dispostas no aplicativo e se elas cumpriram com as suas devidas responsabilidades. Pode-se citar como exemplo a opção de obter o histórico de um sensor isolado selecionando um intervalo de datas pelo usuário. Também é possível configurar valores de máximo e mínimo para cada sensor e no momento que uma variação fora desta faixa for detectada, a aplicação irá demonstrar de forma diferenciada o sensor em questão.



Os testes em outros sistemas operacionais de *smartphone*, como o Android, não foram realizados, embora a aplicação tenha compatibilidade devido a tecnologia empregada, o *framework* React Native.

## 6.2 RESULTADOS E DISCUSSÕES

Finalizadas as etapas de desenvolvimento e de testes do protótipo, tornou-se possível realizar a análise e a observação dos resultados obtidos, dando assim margem para avaliar a viabilidade da aplicação.

O microcontrolador utilizado no projeto, mais especificamente o Arduino modelo Mega 2560, teve como responsabilidade centralizar os valores das grandezas físicas capturadas através dos sensores. Esse dispositivo teve a proficiência de gerenciar os dados e enviá-los para o servidor conforme o esperado.

O módulo capaz de viabilizar a conexão do Arduino com a Internet, o ESP8266, com suas características de comunicação foi utilizado no protótipo. Durante o desenvolvimento do projeto, encontrou-se dificuldades para programar a comunicação entre este dispositivo e o servidor. Por meio de extensas pesquisas e testes foi possível encontrar uma solução utilizando a biblioteca correta a ser empregada.

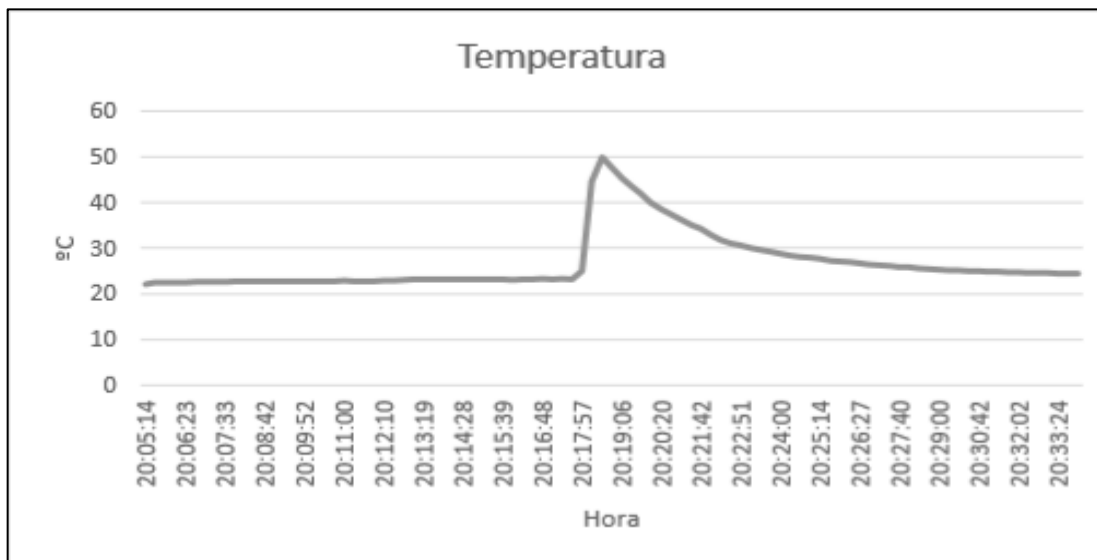
O serviço de processamento e armazenamento gratuito disponibilizado pelo servidor *000webhost* contemplou perfeitamente todas suas funcionalidades estabelecidas e com disponibilidade de vinte e quatro horas por dia, sete dias da semana, provendo assim um funcionamento ininterrupto do sistema. Essa etapa de comunicação entre os dispositivos (aplicação móvel e sensores) depende diretamente da conexão com a Internet, logo podem ocorrer problemas inesperados. No cenário em que esta conexão é interrompida o sistema não tem a autonomia de concluir o seu processo completo, dando uma margem para que os dados coletados acabem se perdendo neste período de ociosidade do acesso à Internet.

A fim de analisar os sensores do protótipo, foram extraídos da base de dados os valores salvos ao decorrer dos testes. Posteriormente foi importado esse arquivo para a ferramenta Excel com a finalidade de produzir um gráfico para auxiliar na visualização dos valores aferidos dentro de um período.

Os valores de temperatura e umidade relativa do ar foram os que se apresentaram dentro do esperado nos testes realizados e respeitou a variação de

forma dinâmica e rápida. Os resultados obtidos permitiram concluir que o sensor de temperatura e umidade obteve êxito no protótipo desenvolvido. A figura 42 demonstra os valores obtidos através do sensor de temperatura, nota-se que houve uma variação pois foi o momento em que foi estressado o sensor com o uma fonte de ar quente.

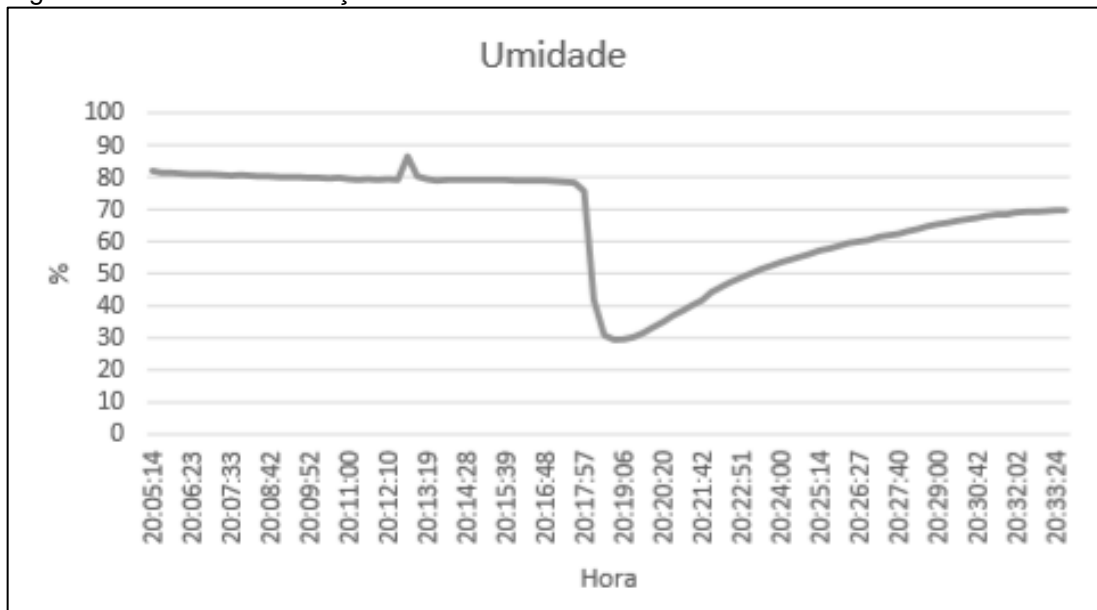
Figura 42 – Gráfico da variação de temperatura



Fonte: Do autor

A variação dos valores aferidos a respeito da umidade relativa do ar pode ser observada na figura 43.

Figura 43 – Gráfico da variação da umidade

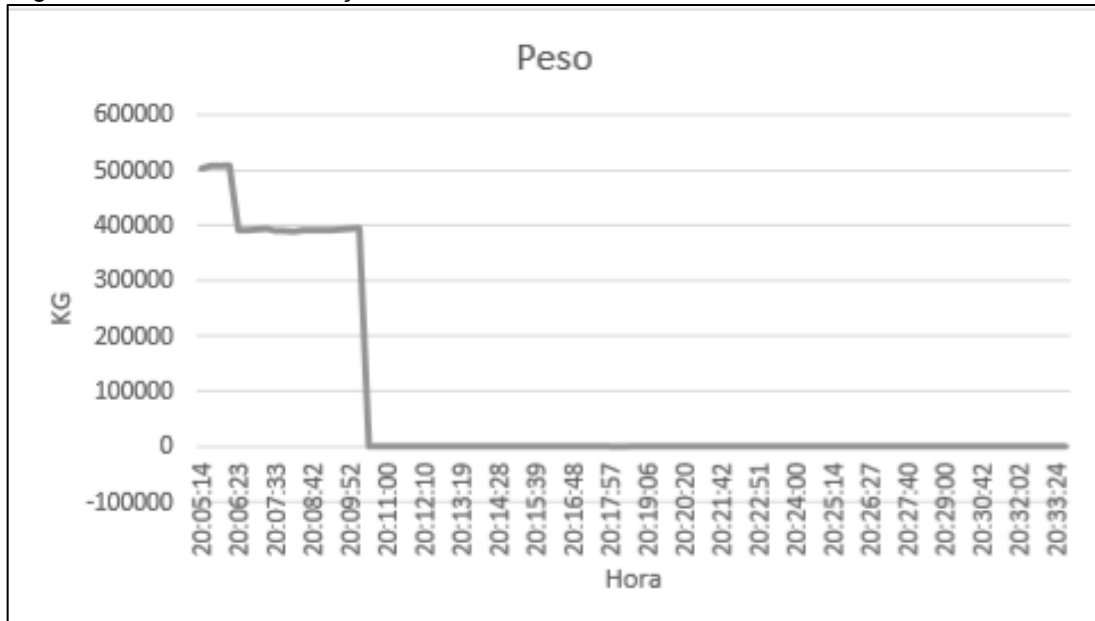


Fonte: Do autor

O sensor de peso, *Strain Gauge*, embora em poucos momentos aferiu de forma precisa o valor na qual foi submetido o peso, na sua grande maioria demonstrou

valores aleatórios não condizentes (figura 44). Realizado os testes de forma isolada com este sensor, levantou-se a hipótese onde as conexões (fios) estavam interferindo em seu funcionamento, existindo algum mal contato no sensor.

Figura 44 – Gráfico da variação da unidade

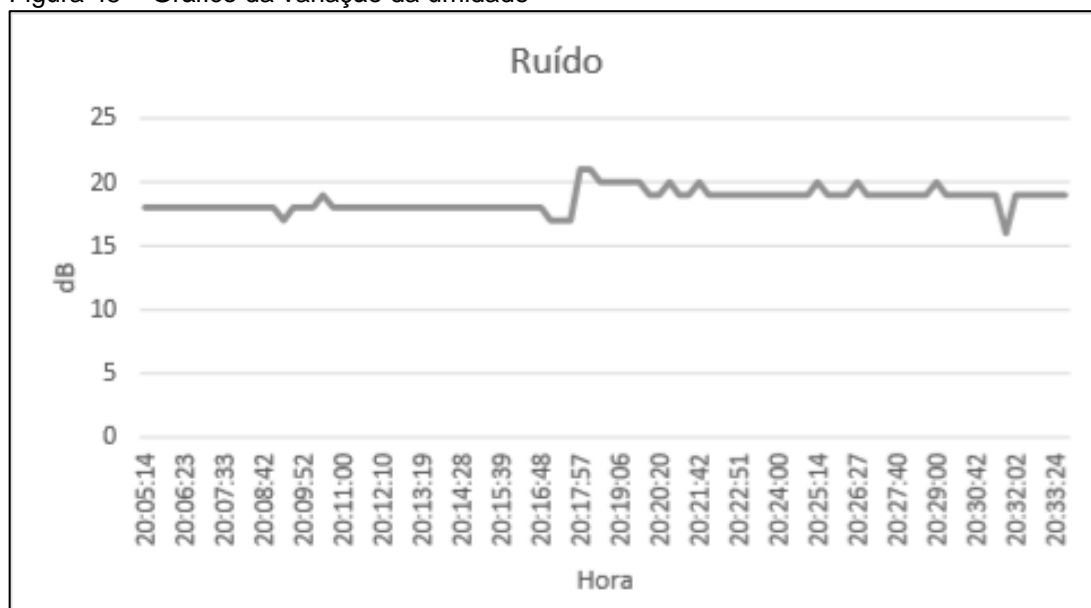


Fonte: Do autor

No protótipo desenvolvido foi empregado apenas um módulo para a aferição do peso da colmeia, embora grande parte do material disponibilizado na Internet demonstra o funcionamento de duas células do *Strain Gauge* atuando como uma ponte. Logo levantou-se algumas hipóteses para este seu comportamento inesperado como: as conexões do sensor estarem com defeitos, a utilização de apenas um módulo dificulta as leituras da grandeza física, do que quando utilizadas duas células e que o módulo conversor (HX711) pode estar com algum possível um defeito.

Referente ao KY-038 (figura 45), embora de fácil manuseio e de baixa complexidade para programar, os resultados obtidos nos testes não se demonstraram satisfatório no momento que o valor aferido referente a vibração sonora não se alterava em nenhum dos cenários, tanto com pouco ruído quanto com muito.

Figura 45 – Gráfico da variação da umidade



Fonte: Do autor

O trabalho desenvolvido por Saraiva (2003) tinha o intuito de centralizar informações de diversos formatos de arquivos, como vídeo, imagens e textos. O projeto, denominado como WebBee, consta com um módulo de sensoriamento capaz de realizar leituras de dados como temperatura, umidade e fluxo de entrada e saída das abelhas da colmeia. Comparado com o protótipo desenvolvido observa-se que o mesmo não consta com um módulo de sensor capaz de verificar este fluxo de entrada e saída da colmeia. Uma nova versão do projeto poderia contemplar esta possibilidade deixando mais completo e robusto.

Outras características do protótipo apresentado, neste trabalho dizem respeito à ausência de medições de gases poluentes e o fato dos sensores aplicados apresentarem conexão por meio de fio para efetuar a comunicação até o micro controlador. Estes pontos foram abordados por Murphy et al. (2015), onde apresentou um sistema contendo uma rede de sensores sem fio para coletar uma variedade de dados de uma colmeia afim de descrever com exatidão as condições internas e a atividade das colônias. Em seu trabalho as variáveis aferidas foram: dióxido de carbono, oxigênio, gases poluentes, temperatura, umidade relativa e aceleração.

O trabalho de Kandepi (2015) também buscou monitorar variáveis, como a temperatura e o ruído no interior de uma colmeia e tendo como seu diferencial a forma de obter a energia para sustentar seu sistema onde os sensores são alimentados através de energia solar. Neste projeto proposto a energia para alimentar o protótipo

depende de uma rede elétrica e também a possibilidade de uma bateria com capacidade para 5V a 12V para suprir a necessidade do protótipo.

Quando comparado a forma de exibir e manipular os dados, destaca-se a diferença do projeto proposto frente ao desenvolvido por Kandepi. No vigente trabalho os dados são salvos em um servidor na nuvem e posteriormente disponibilizado em uma aplicação móvel. Kandepi exibiu as informações em um dispositivo com um *display* de LCD em tempo real, porém o seu sistema não suporta nenhum tipo de armazenamento referente os dados coletados.

O projeto proposto contempla sensores que foram utilizados na maioria dos protótipos já desenvolvidos.

## 7 CONCLUSÃO

Com a finalização do trabalho em questão adquiriu-se além de conhecimentos referentes as tecnologias que foram empregadas para o desenvolvimento do protótipo, possibilitou conhecer mais a respeito do manejo de abelhas, como fatores externos influenciam na produção do mel e seus derivados além de aprender mais a respeito da apicultura.

Esse estudo do comportamento das abelhas possibilitou a análise de quais variáveis seriam aferidas afim de maximizar a produção de mel em uma colmeia. Constatou-se que as grandezas físicas mais relevantes a serem monitoradas deveriam ser: temperatura, umidade relativa do ar, ruído e peso. Para isso, o sistema dispôs de sensores capazes de mensurar os valores dessas grandezas que são transferidos para uma aplicação móvel. O aplicativo também permite visualizar um histórico de um determinado período estipulado pelo apicultor. Outra finalidade que este possui é definir valores máximos e mínimos para cada sensor tendo um *feedback* visual, caso o valor aferido esteja fora deste limite.

Com base nas características que moldam a tecnologia de Internet das Coisas, foi possível integrar e realizar uma comunicação entre os dispositivos do protótipo. Para gerenciar os sensores fez-se o uso do micro controlador Arduino Mega 2560. Os módulos foram acoplados nele possibilitando a realização de leituras dos sensores que posteriormente são enviadas para um servidor na nuvem, tendo como meio a Internet.

Quanto ao desenvolvimento do aplicativo, o maior desafio foi aprender uma nova tecnologia para ser empregado no projeto. Embora o React Native seja um *framework* relativamente novo, possui inúmeros materiais na Internet possibilitando assim um aprendizado rápido e prático.

Com o projeto desenvolvido e tendo uma visão como um todo, observa-se um resultado satisfatório quanto a integração de diversas tecnologias diferentes que foram empregadas, tendo como objetivo final monitorar variáveis em uma colmeia afim de otimizar a produção e diminuir custos. Os sensores de vibrações sonoras e de peso não demonstraram desempenho satisfatório para um monitoramento em um cenário real devido a um possível defeito nos sensores/conexões, prejudicando uma parcela dos resultados almejados com o projeto.

De acordo com o conhecimento adquirido e em seus respectivos resultados diante do projeto, sugere-se como trabalhos futuros:

- a) aplicar o protótipo em um cenário real, inseridos em uma colmeia afim de observar os dados gerados;
- b) avaliar a possibilidade de integrar o sistema com algum modelo de mineração de dados, fazendo o uso de inteligência artificial visando melhorar a produção na apicultura;
- c) verificar tecnologias paralelas a Internet para a comunicação do protótipo, como por exemplo *Bluetooth*;
- d) avaliar outros modelos de sensores de peso e de vibrações sonoras, que tenham capacidade de mensurar os dados de forma mais fidedigna e confiável.

## REFERÊNCIAS

AQEEL, Adnan. **Introduction to Arduino Mega 2560**. 2018. Disponível em: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>. Acesso em 05 out. 2019.

AL-MUTLAQ, Sara. **Getting Started with Load Cells**. Disponível em: <https://learn.sparkfun.com/tutorials/getting-started-with-load-cells>. Acesso em 05 out. 2019.

ALLEN, Sarah; GRAUPERA, Vidal; LUNDRIGAN, Lee. **Desenvolvimento Profissional Multiplataforma para Smartphone, Iphone, Android, Windows Mobile e Blackberry**. Rio de Janeiro: Alta Books, 2012, 280p.

ALVES, Caio A.; MORAIS, Marcos V. B. de. **SENSIBILIDADE DO SENSOR DE INTENSIDADE SONORA PARA DETECÇÃO DE GRANIZO**. REGRAD, UNIVEM/Marília-SP, v. 11, n. 1, p 16-27, agosto de 2018.

ALMEIDA, Gesline Fernandes de. **Fatores que interferem no comportamento enxameatório de abelhas africanizadas**. 208. 120 f. Tese (Doutorado) - Curso de Ciências, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo, Ribeirão Preto, 2008.

AVIA SEMICONDUCTORS. **24-bit Analog-to-Digital Converter (ADC) for Weigh Scales**. Disponível em: [https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711\\_english.pdf](https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf). Acesso em 05 out. 2019.

AMPUDIA, Ricardo. **Brasil lidera número de smartphones conectados na América Latina**. Disponível em: <http://www1.folha.uol.com.br/mercado/2017/09/1917782-brasil-lidera-numero-de-smartphones-conectados-na-america-latina.shtml>. Acesso em: 1 Maio de 2018.

AOSONG ELECTRONICS. **Temperatura and humidity module: AM2302 Product Manual**. Disponível em: [https://img.filipeflop.com/files/download/Datasheet\\_DHT22\\_AM2302.pdf](https://img.filipeflop.com/files/download/Datasheet_DHT22_AM2302.pdf). Acesso em: 05 out. 2018.

ARAÚJO, Regina Borges. (2003) **Computação Ubíqua: Princípios, Tecnologias e Desafios**. In: Simpósio Brasileiro de Redes de Computadores, XXI, Natal. Minicurso: Livro Texto. Natal, RN: UFRN/DIMAp: UnP, 2003. 363 p. p. 45-115.

ARDUINO. **Site Oficial Arduino**, 2018. Disponível em <http://arduino.cc>. Acesso em: 2 out. 2018.

ARDUINO. **Arduino**. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 05 jun. 2018.

\_\_\_\_\_. **Arduino Mega 2560**. 2019. Disponível em: <https://store.arduino.cc/arduino->



mega-2560-rev3. Acesso em: 14 out. 2018.

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The Internet of Things: A survey. **Computer Networks**, [s.l.], v. 54, n. 15, p.2787-2805, out. 2010. Elsevier BV. <http://dx.doi.org/10.1016/j.comnet.2010.05.010>.

BENCSIK, M.; BENCSIK, J.; BAXTER, M.; LUCIAN, A.; ROMIEU, J.; MILLET, M. BERALDO, R.M. **Apicultura Orgânica - Dossiê Técnico**. São Paulo, 2011.

BROWN, M. J.; PAXTON, R. J. **The conservation of bees: a global perspective**. Apidologie, EDP Sciences, v. 40, n. 3, p. 410–416, 2009.

\_\_\_\_\_. **Como funcionam as aplicações web**. 2012. Disponível em: <https://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>. Acesso em: 05 out. 2019

CÂMARA, M. **Bluetooth: O que é e como funciona**. Disponível em < <https://www.techtudo.com.br/artigos/noticia/2012/01/bluetooth-o-que-e-e-como-funciona.html>>. Acesso em 17 nov. 2018.

CAMARGO ,Ricardo Costa Rodrigues de. et al. **Produção de mel**. Teresina: Embrapa Meio Norte, 2002. 138 p. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/bitstream/doc/66838/1/Doc78.pdf>>. Acesso em: 26 de set. de 2018.

DATASHEETCAFE. **KY-038 PDF Arduino – Microphone Sound Sensor Module**. 2016. Disponível em: <<http://www.datasheetcafe.com/ky-038-pdf-arduino/>>. Acesso em: 05 out. 2019.

D´ALMEIDA, J. E. Carvalho. **Abelhas: manual prático do apicultor**. 4. ed. Lisboa: Livraria Popular Francisco Franco, 1983. 158 p.

DEIDMAR, Gabriel Lima Caitano; SOBREIRA, Demóstenes da Silva; LIMA, Welton Dias de. Internet das coisas na Educação. **Revista Tecnologias em Projeção**, Brasília, v. 8, n. 2, p.67-78, dez. 2017.

DOMINGOS, Hérica Girlane Tertulino; GONÇALVES, Lionel Segui. Termorregulação de abelhas com ênfase em Apis mellifera. **Acta Veterinaria Brasilica**, v. 8, n. 3, p.151-154, 2014.

DUARTE, I. **Pt100, pt1000 ou NTC – qual ´eo elemento de medição correto?**. 28 de Março de 2017. Disponível em < <https://blog.wika.com.br/know-how/pt100-pt1000-ou-ntc-qual-e-o-elemento-de-medicao-correto/>>. Acesso em 05 out. 2019

DUTRA, Thiago Fernandes Silva. **BeeHiveior: Sistema de monitoramento e controle de colmeias de produção apícola**. 2016. 81 f. Dissertação (Mestrado) - Curso de Engenharia de Software, Universidade Federal do Rio Grande do Norte, Natal, 2016. Disponível em: <<https://repositorio.ufrn.br/jspui/bitstream/123456789/21775/1/ThiagoFernandesSilvaDutra DISSERT.pdf>>. Acesso em: 10 out. 2018.

EISENMAN, B. **Learning React Native: Building Native Mobile Apps with JavaScript**. O'Reilly Media, Inc. 3 de dez de 2015 - 272 páginas.

EMBRAPA. **Criação de abelhas**: apicultura. Brasília: Embrapa Informação Tecnológica, 2007. Autores: Embrapa Informação Tecnológica/Embrapa Meio-Norte.

ESPRESSIF SYSTEMS. **ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266**. 2013. Disponível em: [https://cdn-shop.adafruit.com/datasheets/ESP8266\\_Specifications\\_English.pdf](https://cdn-shop.adafruit.com/datasheets/ESP8266_Specifications_English.pdf). Acesso em: 07 out. 2019.

EVANS, M; NOBLE, J; HOCHENBAUM, J; **Arduino em ação**. São Paulo: Novatec, 2013.

FERRARI, S. et al. Monitoring of swarming sounds in bee hives for early detection of the swarming period. **Computers And Electronics In Agriculture**, [s.l.], v. 64, n. 1, p.72-77, nov. 2008. Elsevier BV. <http://dx.doi.org/10.1016/j.compag.2008.05.010>.

FERNÁNDEZ, P.; SANTANA, J. M.; ORTEGA, S.; TRUJILLO, A.; SUÁREZ, J. P.; Identification of the honey bee swarming process by analysing the time course of hive vibrations. **Computers and electronics in agriculture**, Elsevier, v. 76, n. 1, p. 44–50, 2011.

FILIPEFLOP. **Sensor de Temperatura e Umidade HDC1080 Alta Precisão**. Disponível em <<https://www.filipeflop.com/produto/sensor-de-temperatura-e-umidade-hdc1080-alta-precisao/>>. Acesso em 17 nov. 2018

FINGERPOINTENG. **ESP 8266 ESP-01 WI-FI MODULE**. 2018. Disponível em: <https://www.fingerpointengg.com/product/esp-8266-esp-01-wifi-module/>. Acesso em: 29 jan. 2019.

FITZGERALD, D. W.; MURPHY, F. E.; WRIGHT, W. M.; WHELAN, P. M.; POPOVICI, E. M. Design and development of a smart weighing scale for beehive monitoring. In: IEEE. **Signals and Systems Conference (ISSC)**, 2015 26th Irish. [S.l.], 2015. p. 1–6.

GOOGLE. **Discover Android**. Disponível em: <<http://www.android.com/about/>>. Acesso em: 20 nov. 2018.

GOK, Nizametin; KHANNA, Nitin. **Building Hybrid Apps with Java and JavaScript**. Sebastopol,CA: O'Reilly, 2013. 156p.

INSTITUTO NCB. **Como utilizar o sensor de temperatura e umidade relativa DHT22 com o NodeMCU (MIC162)**. Disponível em: <http://www.newtoncbraga.com.br/index.php/microcontrolador/143-tecnologia/14349-como-utilizar-o-sensor-de-temperatura-e-umidade-relativa-dht22-com-o-nodemcu-mic162>. Acesso em: 26 out. 2019.

JSON. **Introdução ao JSON**. Disponível em: <https://www.json.org/json-pt.html>.

Acesso em 06 out. 2019.

KANDEPI, M. **Bee hive temperature and sound monitor**. Dissertação (Mestrado). Graduate school of Cornell University, 2015.

LANDIM, W. **O que é Wi-Fi?** Disponível em < <https://www.tecmundo.com.br/wi-fi/197-o-que-e-wi-fi.htm>>. Acesso em 17 nov 2018.

LECHETA, Ricardo R. **Desenvolvendo para iPhone e iPad**: aprenda a desenvolver aplicações utilizando o iOS SDK. 3. ed. São Paulo: Novatec, 2014. 624 p.

LECHETA, Ricardo R. **Google Android**: aprenda a criar aplicações para dispositivos móveis com o Android SDK. 4. ed. rev. e ampl. São Paulo: Novatec, 2015. 1016 p.

LEE, Valentino. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. 1ª edição, 2005, Pearson Education do Brasil, São Paulo.

LEE, Wei-Meng. **Beginning Android 4 Application Development**. 1ª Edição. 2012, John Wiley & Sons, Inc., Indianapolis, Indiana.

LEWIS, T. W. **Beehive monitor**. Relatório de Graduação. California Polytechnic State University, Computer Engineering Department. San Luis Obispo, California. 2014.

LIMA, Nelson Mello de. **Abelhas e mel**: criação e extração. Rio de Janeiro: Tecnoprint, 1979. 149 p.

MARTINEZ, Omar Arvey; SOARES, Ademilson Espencer Egea. Melhoramento genético na apicultura comercial para produção da própolis. **Rev. bras. saúde prod. anim.**, Salvador, v. 13, n. 4, p. 982-990, Dec. 2012. Available from <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1519-99402012000400006&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1519-99402012000400006&lng=en&nrm=iso)>. access on 05 Nov. 2019. <http://dx.doi.org/10.1590/S1519-99402012000400006>.

MASTERWALKER ELETRONIC SHOP. **Sensor (Detector) de Som - KY-038**. Disponível em: <https://www.masterwalkershop.com.br/sensor-detector-de-som-ky-038>. Acesso em: 06 out. 2018.

MDN. **O que é JavaScript?**. Disponível em [https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\\_steps/O\\_que\\_e\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript). Acesso em 06 out 2019.

MELLIS, D.; BANZI, M.; CUARTIELLES, D.; IGOE, T. **Arduino**: An open electronics prototyping. In: CHI 2007 CONFERENCE. Alt.chi 2007. San Jose (CA – EUA), 2007.

MEZQUIDA, D. A.; MARTÍNEZ, J. L. Platform for bee-hives monitoring based on sound analysis. a perpetual warehouse for swarm daily activity. **Spanish Journal of Agricultural Research**, v. 7, n. 4, p. 824–828, 2009.

MONK, Simon. **30 Projetos com Arduino**. 2. ed. Porto Alegre: Bookman, 2014. 214 p.

MONTAN, J. S.; SANTOS, M. C. P. AVALIAÇÃO DE PLATAFORMAS HÍBRIDAS PARA DESENVOLVIMENTO DE APLICAÇÕES PARA O ANDROID-EVALUATION OF HYBRID ANDROID DEVELOPMENT PLATFORMS. **Multiverso: Revista Eletrônica do Campus Juiz de Fora-IF Sudeste MG**, v. 2, n. 2, p. 116-127, 2017.

MURPHY, F. E.; MAGNO, M.; O'LEARY, L.; TROY, K.; WHELAN, P.; POPOVICI, E. M. Big brother for bees (3b)—energy neutral platform for remote monitoring of beehive imagery and sound. In: IEEE. **Advances in Sensors and Interfaces (IWASI), 2015 6th IEEE International Workshop on**. [S.l.], 2015. p. 106–111.

MUXFELDT, Hugo. **Apicultura para todos**. 6 ed. Porto Alegre: Sulina, 1987. 242 p.

MYSQL. **Why SQL?**. Disponível em: <https://www.mysql.com/why-mysql/>. Acesso em: 06 out. 2019.

ODDWIRES. **HC-05 BLUETOOTH TRANSCEIVER MODULE RS232 / TTL MASTER / SLAVE MODULE**. 2019. Disponível em: <https://www.oddwires.com/hc-05-bluetooth-transceiver-module-rs232-ttl-master-slave-module/>. Acesso em: 05 out. 2019.

OLIVEIRA, F.A.M.A.; SEABRA, M.A.M. **Apicultura em Imbassai: viabilidade econômica**. Mata de São João, 2006. Disponível em: <<http://www.institutoimbassai.org.br/arquivos/Projetos/ApiculturaRelatorio.pdf>>. Acesso em: 16 set. 2018.

ORDONEZ, E., D., M.. at al. **Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs)**. Bless Gráfica e Editora Ltda. 2003.

ORDONEZ, E., D., M.; Penteado, C., G.; Silva, A., C., R.. **“Microcontroladores e FPGAs Aplicações em Automação”**. Novatec Editora. 2006.

PEREIRA, F.M., et al. **Produção de mel**. Teresina, 2003. Disponível em: <<http://sistemasdeproducao.cnptia.embrapa.br/FontesHTML/Mel/SPMel/index.htm>>. Acesso em: 20 set. 2018.

PÉREZ, Nicolás et al. Continuous monitoring of beehives' sound for environmental pollution control. **Ecological Engineering**, [s.l.], v. 90, p.326-330, maio 2016. Elsevier BV. <http://dx.doi.org/10.1016/j.ecoleng.2016.01.082>.

PILAR, Carina Ponzoni. **AVALIAÇÃO DAS ARQUITETURAS DE DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS**. 2013. 89 f. TCC (Graduação) - Curso de Bacharelado em Sistemas de Informação, Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul, Caxias do Sul, 2013.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. **Anais do EATI**, ano, v. 4, p. 72-79, 2014.

RABELO, E. **Styled-components: Padrões em produção**. Disponível em: <https://medium.com/nossa-coletividade/styled-components-padr%C3%B5es-em-produ%C3%A7%C3%A3o-4958e91d4d92> . Acesso em 06 out. 2019.

RAMPARANY, F.; MARQUEZ, F. G.; SORIANO, J.; ELSALEH, T. **Handling smart** RODRIGUES, W. C. Fatores que influenciam no desenvolvimento dos insetos. **Info Insetos**, v. 1, n. 4, p. 14–17, 2004.

RAY, John. **Sams Teach Yourself iPhone Application Development in 24 Hours**. 2ª Edição, 2011, Pearson Education, Inc.

ROCHA, A.M., FINZI, R. M. N. **Introdução à Arquitetura Apple iOS**. Universidade Federal de Goiás, Catalão. 2014. Disponível em: [http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011\\_submission\\_58.pdf](http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_58.pdf). Acesso em: 10 nov. 2018.

MDN WEB DOCS. **Uma visão geral do HTTP**. 2018. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>. Acesso em: 05 out. 2019.

SANTUCCI, G. The Internet of things: Between the revolution of the Internet and the metamorphosis of objects. **Vision and Challenges for Realising the Internet of Things**, p.11–24, 2010.

SARAIVA, A.M. WebBee – a Web-based Information Network on Bees. **Revista de Engenharia de Computação e Sistemas Digitais**, São Paulo, v. 1, nov. 2003. P. 77-86.

SCHIRMER, Lenhart Robert. **Abelhas ecológicas**. São Paulo: Nobel, 1985. 218 p.

SERVELIN, Augusto Vítório. **Um aplicativo computacional na disciplina de apicultura**. 48 f. Dissertação (Mestrado em Ciências) Programa de Pós-Graduação em Educação Agrícola, Área de Concentração em Educação Agrícola. 2008.

SOUZA, Fábio. **Arduino MEGA 2560**. 2014. Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 03 nov. 2018.

TELECO. **Mercado mundial de smartphones**. Disponível em: [http://www.teleco.com.br/sist\\_operacional.asp](http://www.teleco.com.br/sist_operacional.asp). Acesso em: 6 nov. 2018.

TEXAS INSTRUMENTS. **HDC1080 Low Power, High Accuracy Digital Humidity Sensor with Temperature Sensor**. 2017. Disponível em: <http://www.ti.com/lit/ds/symlink/hdc1080.pdf>. Acesso em: 05 out. 2019.

TEXAS INSTRUMENTS. **LM35 Precision Centigrade Temperature Sensors**. 2017. Disponível em: <http://www.ti.com/lit/ds/symlink/lm35.pdf>. Acesso em: 05 out. 2019.

VIDAL, Vitor. **Balança digital com arduino: aprenda a usar a célula de carga**. 2018. Disponível em: <https://blog.eletrogate.com/balanca-digital-com-arduino-aprenda-a-usar-a-celula-de-carga/>. Acesso em 05 out. 2019.

VIEIRA, Márcio Infante. **Criar abelhas é lucro certo**: Manual prático. São Paulo: Nobel, 1989. 175 p.

WEISER, M. (1991), "**The Computer for the 21st Century**", Scientific American, vol.265,no.3,Setembro., pp.94- 104. Apud {1}.

WIESE, Helmuth. **A nova apicultura**. 6 ed. Porto Alegre: Agropecuária, 1985. 493 p.

WILSON, T. **Como funcionam as abelhas**. [S.l.], 2001.

WILKEN, Jeremy. **Ionic in Action**. Estados Unidos: Manning Early Access Program, 2015. 325 p.

ZACEPINS, A.; STALIDZANS, E.; MEITALOVS, J. Application of information technologies in precision apiculture. In: **Proceedings of the 13th International Conference on Precision Agriculture (ICPA 2012)**. [S.l.: s.n.], 2012.

## **APÊNDICE(S)**

## Monitoramento de Colmeia por meio de Internet das Coisas conectado a uma Aplicação Mobile

Tiago Behenck dos Santos, Paulo João Martins<sup>2</sup>

<sup>1</sup>Acadêmico do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma – SC

<sup>2</sup>Professor do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma - SC

tiago.behenck.santos@gmail.com, pjm@unesc.net

**Abstract.** *The use of technologies in various areas of agricultural production is a Brazilian reality. In the field of beekeeping the scenario does not differ, large producers have access to technological tools seeking to optimize productivity. But access to technological means is not a unanimous reality in agriculture. Given this context, the present work aims to research and develop a low cost solution for monitoring the temperature, relative humidity, weight and noise variables in a hive. To this end, a prototype was developed, applying the Internet of Things concepts and using the Arduino prototyping board integrated with specific sensors for each monitored variable. Data is collected and sent to a cloud server. Subsequently, to manage this information, a mobile application was developed using the React Native framework, where the beekeeper has the possibility to view and also perform a search within the desired date range regarding the collected data.*

**Resumo.** *O emprego de tecnologias em diversas áreas da produção agrícola é uma realidade brasileira. No ramo da apicultura o cenário não se difere, grandes produtores têm acesso a ferramentas tecnológicas buscando otimizar a produtividade. Porém o acesso a meios tecnológicos não é uma realidade unânime na agricultura. Diante deste contexto, o presente trabalho objetiva a pesquisa e o desenvolvimento de uma solução de baixo custo para o monitoramento das variáveis de temperatura, umidade relativa do ar, peso e ruído em uma colmeia. Para tal foi desenvolvido um protótipo, aplicando os conceitos de Internet da Coisas e utilizando-se a placa de prototipação Arduino integrado a sensores específicos para cada variável monitorada. Os dados são coletados e enviados para um servidor na nuvem. Posteriormente, para o gerenciamento dessas informações, foi realizado o desenvolvimento de uma aplicação mobile, utilizando-se do framework React Native, onde o apicultor tem a possibilidade de visualizar e também efetuar uma pesquisa dentro de intervalo de data desejado a respeito dos dados coletado.*



## 1. Introdução

Entende-se por cultura ou exploração das abelhas, ou ainda apicultura, o ramo de agricultura que tem por fim explorar esses insetos e tirar deles o máximo e o melhor rendimento possível (D'ALMEIDA, 1983). As abelhas têm um papel fundamental no ecossistema. Conforme Lima (1979), seriam grandes as dificuldades de alimentação da humanidade sem a existência das abelhas, pois elas são os principais insetos polinizadores que existem e sem elas milhares de espécies de plantas teriam deixado de existir. Além da sua função ecológica, as abelhas e seus produtos demonstram um grande potencial econômico.

A colheita de mel se trata de uma atividade que requer esforço físico acentuado causando um desgaste para o apicultor, uma vez que o peso das colmeias cheias é considerável (CAMARGO et al., 2002). A forma de verificação das colmeias por vezes pode se mostrar invasiva, prejudicando a produção de mel. Tratando-se de um grande produtor, pode acarretar prejuízos, além do custo da mão de obra qualificada para exercer determinada função. Visando estes pontos demonstra-se viável o desenvolvimento de um sistema que auxilie nas averiguações periódicas de algumas variáveis (umidade, temperatura, peso e ruído) simplificando o processo de aferição manual que seria apenas a retirada do mel.

O governo de Santa Catarina já monitora algumas colmeias, aferindo temperatura e peso, por meio de um sistema desenvolvido pela EPAGRI/CIRAM. Porém o acesso a este tipo de equipamento é restrito devido aos seus altos custos.

Empregando o conceito atual de Internet das Coisas, também conhecida pela Internet of Things (IoT), demonstra uma enorme aplicabilidade para o desenvolvimento do protótipo. O sistema pode se integrar a uma plataforma, seja ela um aplicativo ou um site, de uma forma dinâmica, trazendo os resultados obtidos pelos sensores de maneira simples e eficaz ao usuário. Utilizando sensores de peso, que permita fazer a pesagem de uma “casa de mel”, sem a necessidade de aferir de forma manual, diminui o desgaste físico do apicultor e também reduz o estresse das abelhas.

Se tratando de prototipação, existe o sistema embarcado denominado Arduino, que se trata de uma plataforma de prototipagem eletrônica open source (ARDUINO, 2019, tradução nossa). Se baseia em hardware e software flexíveis e aplicado como ferramenta principal para o processamento e gerenciamento dos sensores. Sua aplicabilidade e integração com inúmeras opções de programação aliado ao seu custo acessível. Com todos os dados coletados da colmeia em questão, pretende-se desenvolver um aplicativo com o intuito de exibir estas informações. Planejou-se criar um aplicativo mobile devido a popularidade do smartphone, contudo há alguns pontos a serem observados, como por exemplo, a existência de colmeias em um terreno isolado, onde inexistente o sinal de telefone ou Internet móvel.

Levando em consideração a importância das abelhas, no contexto da natureza e indo além das questões ambientais, como por exemplo, o quesito econômico onde os pequenos e os grandes produtores encontram dificuldades para obter um monitoramento de precisão que futuramente resulte em uma produção maior de mel, por exemplo. Com esta situação, pretende-se desenvolver um sistema que tem a capacidade de monitorar variáveis (peso, umidade, ruído e temperatura) por meio de sensores, ligados a um aplicativo utilizando Internet das Coisas e exibindo os dados de maneira fácil para o produtor.

Com este sistema há possibilidade de reduzir os custos a longo prazo de extração de mel, além de ter um maior controle quanto a saúde da colmeia, bem como ter dados confiáveis para um melhor planejamento e controle da produção.

## 2. Microcontroladores e Sensores

Os microcontroladores possuem características de um sistema computacional completo em um único chip e são encontrados em uma grande variedade de dispositivos. Possuem internamente processador, memória para armazenar dados do tipo RAM (Random Access Memory), memória para armazenamento de programa do tipo EEPROM (Electrically Erasable Programmable Read-Only Memory), capacidade de entrada e saída e de comunicação, timer gerador de sinais de interrupção, conversor A/D (Analogico Digital), entre outras características específicas de cada fabricante (ORDONEZ et al., 2003).

Um sensor se trata de um dispositivo que detecta e sinaliza as mudanças de condições de um ambiente. Por exemplo, um sensor pode sinalizar a presença ou ausência de um objeto ou material. Um sensor analógico pode aferir grandezas como distância, tamanho e cor, entre outras (ORDONEZ; PENTEADO; SILVA, 2006).

Tendo como estas duas caracterizações o projeto de pesquisa visa empregar ambos no desenvolvimento do protótipo. Exemplificando as opções disponíveis de sensores e da sua utilização, unindo sensores e um microcontrolador afim de aferir as condições de uma colmeia.

### 2.1 Arduino

É uma pequena plataforma de prototipação que possui vários pinos de conexão, permitindo que outros circuitos eletrônicos externos sejam conectados a ele, como por exemplo, motores, relés, sensores, alto-falantes, entre outros. Também possui uma porta Universal Serial Bus (USB), que o permite se conectar a um computador com intuito de realizar o controle do fluxo de dados (MONK, 2014).

De acordo com o site oficial da plataforma (ARDUINO, 2019, tradução nossa), as principais vantagens são: custo, ambiente de programação simples e claro, multiplataforma e software / hardware open source e extensível.

## 3. Aplicações Híbridas

Aplicações híbridas são uma categoria especial de aplicativos da web que ampliam o ambiente de aplicativos por meio de APIs de plataforma nativas disponíveis em um determinado dispositivo (GOK; KHANNA, 2013, tradução nossa). Possuem como finalidade funcionar nos mais diferentes dispositivos, sendo que para diferentes plataformas, será utilizado o mesmo código fonte. Ficam instaladas no dispositivo e podem funcionar de forma independente da conexão com a Internet.

Partindo do princípio de utilizar a mesma aplicação em diferentes plataformas, normalmente as aplicações híbridas são desenvolvidas em linguagens web que são interpretadas pelo browser nativo do sistema (PREZOTTO; BONIATI, 2014).

Dentro da aplicação híbrida há três abordagens para o seu desenvolvimento, são elas: tempo de execução, tradutor de código e Web-para-Nativo.

Na opção tempo de execução o aplicativo é desenvolvido em uma linguagem script (JavaScript, Lua ou Ruby). A ferramenta para o desenvolvimento cria um pacote de instalação do aplicativo com o interpretador da linguagem utilizada e o código script. Em tempo de execução, o interpretador traduz o código script/fonte para código de máquina (nativo) para então ser executado (MONTAN; SANTOS, 2017).

Os tradutores de código se mostram semelhante ao tempo de execução, porém com duas diferenças. A primeira está atrelada a linguagem de desenvolvimento que não se limita a

linguagens script. A segunda está na parte da compilação: o código fonte é compilado para byte-code ou código de máquina (nativo). O byte-code é universal existindo para as principais plataformas (MONTANAN; SANTOS, 2017).

A abordagem Web-para-Nativo (Web-To-Native Wrapper) alia as vantagens das webapps juntamente com aplicações nativas. São utilizadas linguagens de desenvolvimento WEB (HTML, CSS e JavaScript), porém os aplicativos não são executados por um navegador. A aplicação é executada por um container nativo (WebView no Android e UIWebView no iOS) e o acesso ao hardware e recursos do dispositivo é feito por bibliotecas. As principais desvantagens estão ligadas a interface ser distinta da interface nativa da plataforma, já conhecida pelo usuário, e o desempenho, prejudicado pela execução no container (MONTANAN; SANTOS, 2017).

### 3.1 Framework React Native

React Native é um framework JavaScript para desenvolver aplicações móveis em diferentes sistemas operacionais, no caso iOS e Android. É baseada no React, uma biblioteca web desenvolvida pelos engenheiros do Facebook. Em outras palavras, o desenvolvimento web agora pode escrever aplicativos que parecem realmente "nativos", tudo a partir da flexibilidade e poder do JavaScript. (EISENMAN, 2015, tradução nossa). Ele faz uso do Node.js para efetuar os build de código JavaScript, uma tecnologia recente que vem ganhando mercado.

O conceito principal é “aprenda uma vez, escreva em qualquer lugar”. Isso significa um desenvolvedor pode criar um aplicativo web através do ReactJS e também pode criar aplicativos móveis através do React Native sem experiência anterior em desenvolvimento nativo devido a semelhança entre estas duas tecnologias (VIDHALL; HANSSON, 2016, tradução nossa).

Um ponto muito importante desta tecnologia está associada a sua bridge, que seria a ponte entre o javascript e as linguagens nativas (Swift e Java/Kotlin). Os componentes deste framework envolvem o código nativo existente e interagem com as APIs nativas por meio do paradigma declarativo da interface do usuário e do JavaScript (REACT NATIVE, 2019). A figura 6 exemplifica como a bridge atua no framework.

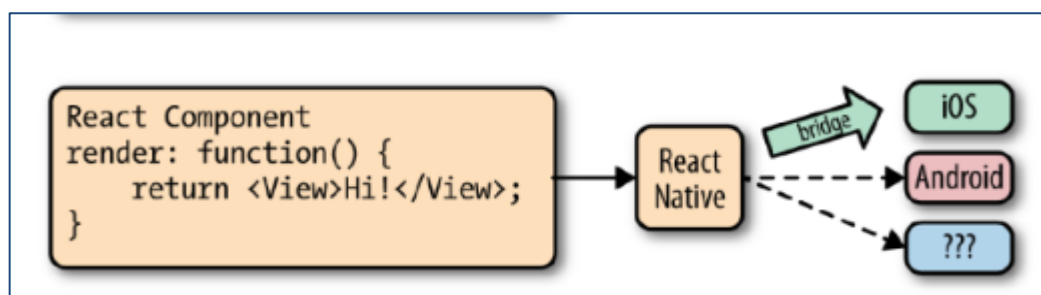


Figura 2. Fluxo de dados do projeto

Esta característica de ponte acaba diferenciando o React Native de outras opções de desenvolvimento de aplicativos híbridos, que geralmente acabam renderizando visualizações na web. O código é traduzido para se adequar à plataforma; portanto, uma tela pode se adequar ao modo de exibição específico do iOS. Devido à camada de abstração, como demonstrado na figura 1, o React Native também pode ter como alvo outras plataformas – se faz necessário a existência de uma bridge.

O fato de o React Native renderizar seus elementos usando a API nativa da plataforma em questão leva a se destacar da maioria das tecnologias existentes no desenvolvimento de aplicativos híbridos, como Cordova ou Ionic. Os métodos existentes para escrever aplicativos móveis usando combinações de JavaScript, HTML e CSS normalmente são renderizados usando a visualização na web. Embora essa abordagem possa funcionar, ela também apresenta desvantagens, especialmente em torno do desempenho. Além disso, essas tecnologias geralmente não têm acesso ao conjunto de ferramentas disponibilizadas pelo sistema operacional. (EISENMAN, 2015, tradução nossa).

#### **4. Metodologia**

O método utilizado para o desenvolvimento deste projeto foi o de pesquisa quantitativa e tecnológica realizada em etapas. Primeiramente foi realizado o levantamento bibliográfico, o qual foi embasado em artigos, livros, publicações e pesquisas na Internet.

As consultas teóricas provenientes da Internet foram extraídas de bases de dados conceituadas, aferindo credibilidade a todo conteúdo descrito. Podem ser citadas as bases: Google Acadêmico; EMBRAPA relacionado a questões técnicas quanto ao cultivo de abelhas; IEEE Xplore; Elsevier e em repositórios de diversas Universidades.

Posteriormente deu-se início a escrita do referencial teórico, abordando questões primordiais para nortear e contextualizar todo o cenário apícola e as ferramentas tecnológicas necessárias que constituem assuntos importantes para analisar e desenvolver o protótipo proposto.

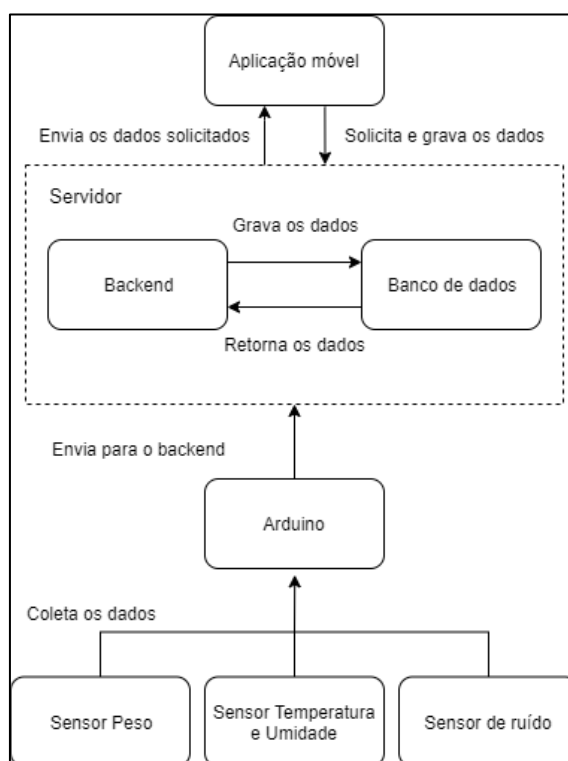
Esta etapa contempla temas como: apicultura e a importância das abelhas na natureza, visando compreender o impacto desse inseto na economia e em seu habitat, além de dar ênfase nas variáveis a serem monitoradas; tecnologias para o desenvolvimento do protótipo, onde engloba Internet das Coisas e as soluções para desenvolvimento mobile com maior destaque para o framework React Native; microcontroladores e sensores para aferir as grandezas físicas ligada a colmeia.

Com base em pesquisas e trabalhos desenvolvidos anteriormente, pode-se observar quais variáveis seriam mais adequadas para monitorar a colmeia.

As etapas consequentes estão ligadas ao desenvolvimento do protótipo em si. Tomando como partida o levantamento dos requisitos necessários, os recursos que melhor se adaptam ao cenário proposto, bem como as vantagens e desvantagens das ferramentas partiu-se à configuração do servidor responsável por armazenar todos os dados captados dos sensores, a montagem do protótipo e testes avaliando a sua precisão, eficácias e possíveis equívocos.

##### **4.1. Fluxo dos dados e modelo de comunicação**

O fluxo dos dados da aplicação apresentado na figura 2, é dividido em quatro elementos principais: aplicação móvel, servidor, Arduino e sensores. Cada grupo tem sua responsabilidade, inter-relacionamentos e co-dependência. O perfeito funcionamento do protótipo só se dará se cada elemento desempenhar sua função no processo.



**Figura 2. Fluxo dos dados dentro da aplicação**

A entrada do fluxo se dá pelos sensores que são responsáveis pela coleta das grandezas físicas, podem ser divididas em peso, temperatura, umidade e vibrações sonoras. Estes módulos estão conectados diretamente ao Arduino, que tem como funcionalidades centralizar os dados e processá-los para então posteriormente transmitir, via Wi-Fi, ao servidor na nuvem. Cada colmeia tem seu conjunto de sensores próprios que fazem a aferição em um intervalo de tempo pré estabelecido, que para fins de testes foi aplicado um intervalo de trinta segundos.

Essa conexão entre o hardware e o servidor é estabelecida através da Internet juntamente com um protocolo de transferência de hipertexto, do inglês HyperText Transfer Protocol (HTTP), servindo-se dos métodos GET e POST, conforme a etapa.

No projeto do monitoramento de colmeias, a transferência de dados do micro controlador ao servidor é feita pelo método POST e a transferência do servidor para o aplicativo pelo método GET.

A responsabilidade do processamento dos dados coletados é inteiramente do servidor. O backend foi desenvolvido em PHP, que se trata de uma linguagem de script open source de uso geral, altamente difundida e principalmente voltada para o desenvolvimento web.

Além de processar os dados, o PHP é responsável por gerenciar a conexão com o banco de dados MySQL. Essa comunicação é dada através de comandos SQL, necessário para retornar as informações de acordo com o que foi solicitado pelo PHP.

Relacionado ao formato dos dados referente em cada transmissão entre as etapas do protótipo, foi padronizado o uso do JavaScript Object Notation (JSON);

Na ponta da arquitetura encontra-se a aplicação móvel, que tem como finalidade exibir os dados aferidos pelos sensores e também estabelecer um valor mínimo e máximo para cada sensor. Caso o dado capturado esteja fora desta margem, a aplicação tem como funcionalidade demonstrar os valores para o usuário.

## 4.2. Montagem da simulação

Realizada a conclusão de todas as etapas referente ao desenvolvimento do projeto, foi possível realizar testes para validar a precisão e acurácia dos sensores, o poder do processamento, a capacidade de armazenamento e a visualização dos dados.

Para o efetivo funcionamento do aplicativo, inicialmente se faz necessário configurar o banco de dados, de forma manual, inserindo as informações referentes a identificação da colmeia a ser monitorada, os parâmetros dos sensores e o cadastro do apicultor. É imprescindível identificar o tipo de sensor através de uma chave única para estabelecer um relacionamento com as demais tabelas, possibilitando a sua identificação na tabela “leituras”, por exemplo. A mesma lógica é aplicada na tabela de colmeias, tornando possível identificar a qual proprietário ela pertence.

Com as devidas tabelas preenchidas é possível inicializar o protótipo a fim de testá-lo, observando como cada componente irá se comportar em diferentes situações estipuladas.

Os testes foram realizados de forma individual em cada sensor, buscando visualizar possíveis alterações nos valores aferidos. Essa atividade de análise do protótipo partiu da conexão do Arduino em uma rede de Wi-Fi com a finalidade de se comunicar e transferir os dados para o servidor.

O sensor de temperatura e umidade foi estressado utilizando um secador de cabelo, regulado para produzir ar quente numa velocidade média, no qual se pode observar que os valores imediatamente responderam às alterações provocados pelo experimento. A temperatura aferida no aplicativo aumentou e houve variação do valor da umidade relativa do ar.

A testagem do sensor de ruído foi feita em dois cenários opostos: um silencioso e outro com barulho. Na exposição do sensor às duas situações, observou que – indiferente da configuração realizada manualmente no potenciômetro responsável pela sensibilidade do microfone – os valores não demonstraram variações. Levantou-se a hipótese de que o sensor empregado estava com alguma avaria, algum defeito nesse sensor em específico e não no modelo em si.

Para analisar a carga, o sensor de peso em si, foi aplicada uma pressão com o intuito de aferir a variação decorrente da deformação da sua resistência. Embora tenha sido calibrado devidamente e testado diferentes formas de desenvolvimento de código, o sensor não demonstrou precisão quando foi sobreposto um objeto de peso conhecido.

O Arduino operou normalmente gerenciando todos os sensores acoplados, realizando as coletas de dados e enviando-os para o servidor em questão, demonstrando nenhum equívoco.

O aplicativo foi testado em um smartphone dispondo do sistema operacional iOS na versão 12.4.1, no modelo iPhone 6S Plus da Apple. Com isso analisou-se todas as funcionalidades dispostas no aplicativo e se elas cumpriram com as suas devidas responsabilidades. Pode-se citar como exemplo a opção de obter o histórico de um sensor isolado selecionando um intervalo de datas pelo usuário. Também é possível configurar valores de máximo e mínimo para cada sensor e no momento que uma variação fora desta faixa for detectada, a aplicação irá demonstrar de forma diferenciada o sensor em questão.

Os testes em outros sistemas operacionais de smartphone, como o Android, não foram realizados, embora a aplicação tenha compatibilidade devido a tecnologia empregada, o framework React Native.

## 4.3. Desenvolvimento da aplicação

A última etapa do desenvolvimento do projeto foi a programação referente a construção do aplicativo. Utilizou-se um framework híbrido, React Native, afim de cobrir a maior parte dos sistemas operacionais embarcados nos smartphones atualmente.

Dentro das características do framework, observou-se o seu desempenho em multiplataforma, comparado com o desenvolvimento nativo. O React Native é uma tecnologia desenvolvida pelo Facebook e de código aberto, conta com uma comunidade entusiasta, facilitando o processo de aprendizagem. O fato de ser multiplataforma tem a vantagem relacionada ao tempo, tornando o desenvolvimento ágil em um curto prazo.

O Expo foi utilizado com o intuito de facilitar o desenvolvimento, com suas ferramentas e abstrações prontas para o uso, uma vez que ele minimiza a complexidade de lidar com questões nativas, além de ser uma forma mais fácil para interagir com o React Native.

Para desenvolver usufruindo dessas ferramentas foi aplicada a linguagem JavaScript. Trata-se de uma linguagem de programação que permite a criação de conteúdos que se atualizam dinamicamente, controlam multimídias, imagens animadas, entre outros recursos (MDN, 2019). O JavaScript é uma linguagem amplamente utilizada, versátil e de fácil aprendizado e vem sendo amplamente utilizada os desenvolvedores

#### 4.4. Funcionamento da aplicação

Após realizadas as etapas de desenvolvimento do protótipo partiu-se para a fase de demonstração do funcionamento do aplicativo. A aplicação é composta por uma série de telas, cada uma com sua finalidade específica, dentre estas, existem telas de consultas dos dados e de manutenção, sendo que todas elas são organizadas pelo conceito de menu, onde a partir deste é possível acessar todos os recursos do aplicativo.

Baseando-se em uma lógica para o acesso cronológico das telas na aplicação, tem-se na primeira página a lista de colmeias cadastradas sendo a principal delas. Selecionando a colmeia desejada é possível acompanhar os últimos valores de cada sensor obtido pelo microcontrolador, com sua respectiva data e hora, e se julgar necessário, também visualizar um histórico detalhado de todas as coletas realizadas. A figura 4 ilustra as funcionalidades mencionadas.

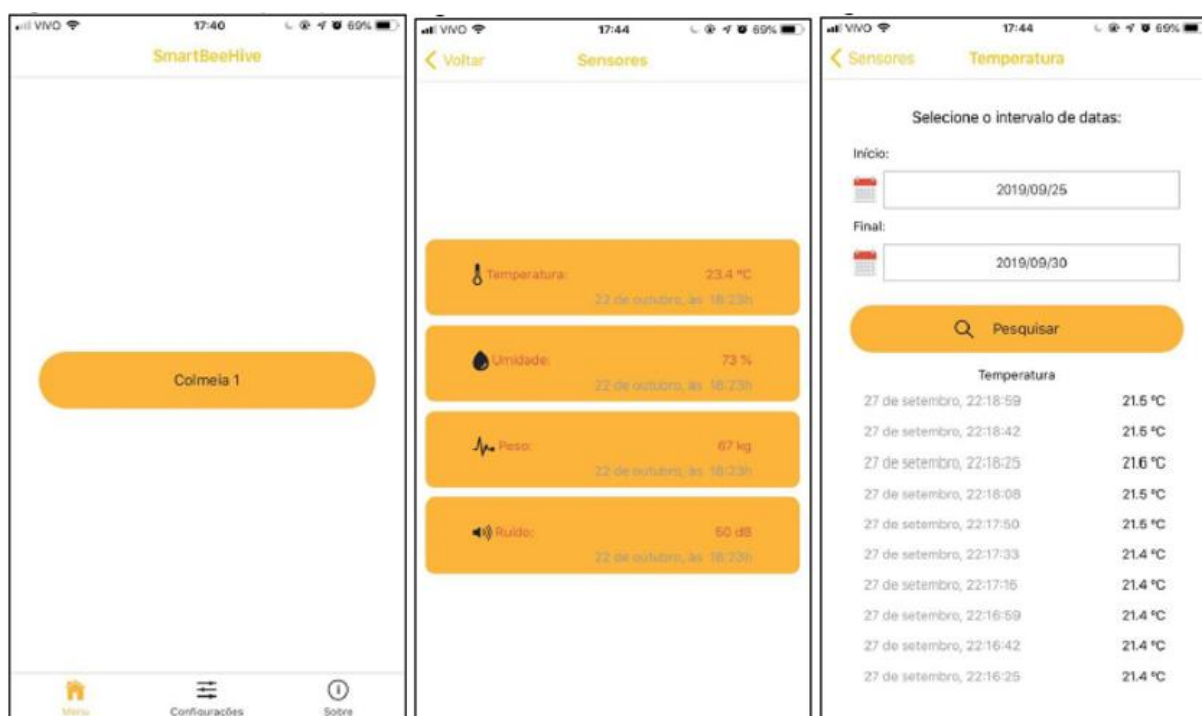


Figura 4. Telas do aplicativo

As demais funcionalidades consistem nas telas de configurações e informações sobre o apicultor. A figura 5 representa as principais.

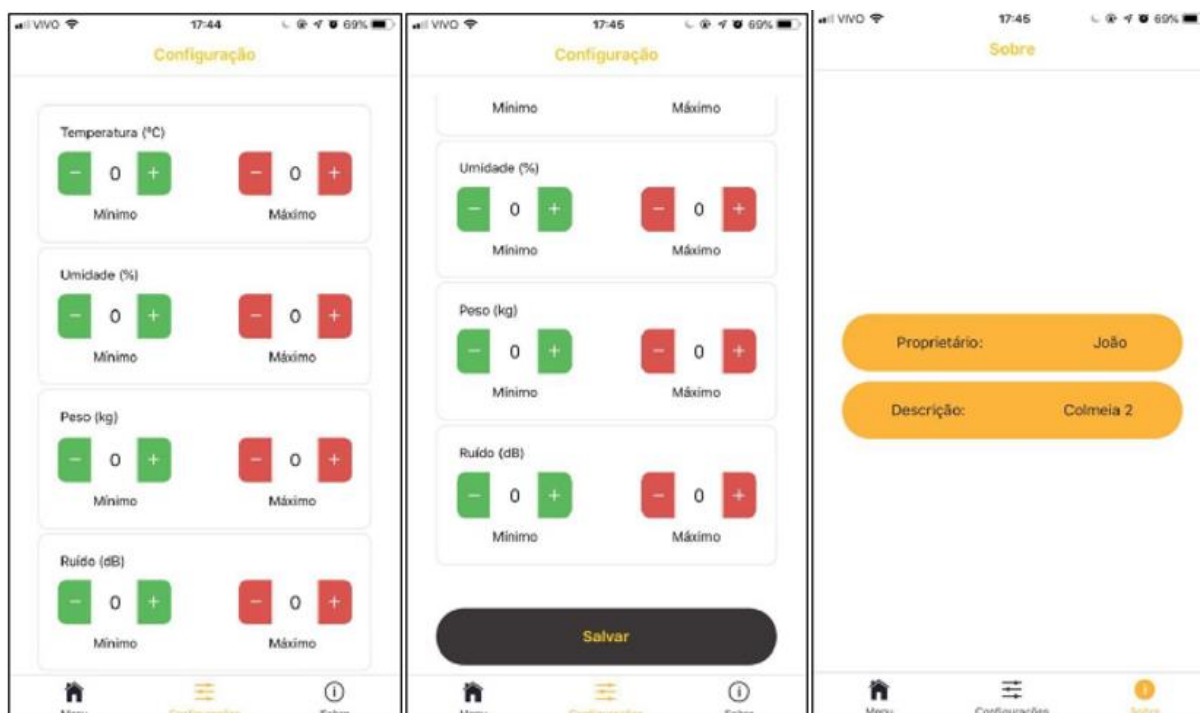


Figura 5. Telas do aplicativo

## 7. Conclusão

Com a finalização do trabalho em questão adquiriu-se além de conhecimentos referentes as tecnologias que foram empregadas para o desenvolvimento do protótipo, possibilitou conhecer mais a respeito do manejo de abelhas, como fatores externos influenciam na produção do mel e seus derivados além de aprender mais a respeito da apicultura.

Esse estudo do comportamento das abelhas possibilitou a análise de quais variáveis seriam aferidas afim de maximizar a produção de mel em uma colmeia. Constatou-se que as grandezas físicas mais relevantes a serem monitoradas deveriam ser: temperatura, umidade relativa do ar, ruído e peso. Para isso, o sistema dispôs de sensores capazes de mensurar os valores dessas grandezas que são transferidos para uma aplicação móvel. O aplicativo também permite visualizar um histórico de um determinado período estipulado pelo apicultor. Outra finalidade que este possui é definir valores máximos e mínimos para cada sensor tendo um feedback visual, caso o valor aferido esteja fora deste limite.

Com base nas características que moldam a tecnologia de Internet das Coisas, foi possível integrar e realizar uma comunicação entre os dispositivos do protótipo. Para gerenciar os sensores fez-se o uso do micro controlador Arduino Mega 2560. Os módulos foram acoplados nele possibilitando a realização de leituras dos sensores que posteriormente são enviadas para um servidor na nuvem, tendo como meio a Internet.

Quanto ao desenvolvimento do aplicativo, o maior desafio foi aprender uma nova tecnologia para ser empregado no projeto. Embora o React Native seja um framework relativamente novo, consta com inúmeros materiais na Internet possibilitando assim um aprendizado rápido e prático.



Com o projeto desenvolvido e tendo uma visão como um todo, observa-se um resultado satisfatório quanto a integração de diversas tecnologias diferentes que foram empregadas, tendo como objetivo final monitorar variáveis em uma colmeia afim de otimizar a produção e diminuir custos quando as tecnologias que existem hoje. Os sensores de vibrações sonoras e de peso não demonstraram desempenho satisfatório para um monitoramento em um cenário real devido a um possível defeito nos sensores/conexões, prejudicando uma parcela dos resultados almejados com o projeto.

## Referências

- ARDUINO. Arduino. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 05 jun. 2018.
- CAMARGO, Ricardo Costa Rodrigues de. et al. Produção de mel. Teresina: Embrapa Meio Norte, 2002. 138 p. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/bitstream/doc/66838/1/Doc78.pdf>>. Acesso em: 26 de set. de 2018.
- D'ALMEIDA, J. E. Carvalho. Abelhas: manual prático do apicultor. 4. ed. Lisboa: Livraria Popular Francisco Franco, 1983. 158 p.
- EISENMAN, B. **Learning React Native: Building Native Mobile Apps with JavaScript**. O'Reilly Media, Inc. 3 de dez de 2015 - 272 páginas.
- LIMA, Nelson Mello de. Abelhas e mel: criação e extração. Rio de Janeiro: Tecnoprint, 1979. 149 p.
- Hansson, N. e Vidhall, T. (2016). Effects on performance and usability for cross-platform application development using react native.
- ORDONEZ, E., D., M.. at al. Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs). Bless Gráfica e Editora Ltda. 2003.
- ORDONEZ, E., D., M.; Penteado, C., G.; Silva, A., C., R.. “Microcontroladores e FPGAs Aplicações em Automação”. Novatec Editora. 2006.
- GOK, Nizamettin; KHANNA, Nitin. Building Hybrid Android Apps: with Java and JavaScript. Sebastopol: O'reilly, 2013. 155 p. Disponível em: <https://doc.lagout.org/programming/Multi-Language/Building%20Hybrid%20Android%20Apps%20with%20Java%20and%20JavaScript%20%5BGok%20%26%20Khanna%202013-08-10%5D.pdf>. Acesso em: 11 nov. 2018.
- MAZUCHETTI, Muriel Rampinelli. ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL. 2015. 79 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2015.
- MONK, Simon. 30 Projetos com Arduino. 2. ed. Porto Alegre: Bookman, 2014. 214 p.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas. 2014. 8 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Universidade Federal de Santa Maria, Frederico Westphalen, 2014. Disponível em: <http://www.eati.info/eati/2014/assets/anais/artigo8.pdf>. Acesso em: 10 nov. 2018.

MDN. O que é JavaScript?. Disponível em [https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First\\_steps/O\\_que\\_e\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript). Acesso em 06 out 2019

MONTAN, J. S.; SANTOS, M. C. P. AVALIAÇÃO DE PLATAFORMAS HÍBRIDAS PARA DESENVOLVIMENTO DE APLICAÇÕES PARA O ANDROID-EVALUATION OF HYBRID ANDROID DEVELOPMENT PLATFORMS. Multiverso: Revista Eletrônica do Campus Juiz de Fora-IF Sudeste MG, v. 2, n. 2, p. 116-127, 2017.